



[Home](#)



OS-9[®] Device Descriptor and Configuration Module Reference

Version 4.7



RadiSys
THE POWER OF WE

www.radisys.com

Revision A • July 2006

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.

July 2006
Copyright ©2006 by RadiSys Corporation
All rights reserved.

EPC and RadiSys are registered trademarks of RadiSys Corporation. ASM, Brahma, DAI, DAQ, MultiPro, SAIB, Spirit, and ValuePro are trademarks of RadiSys Corporation.

DAVID, MAUI, OS-9, OS-9000, and SoftStax are registered trademarks of RadiSys Corporation. FasTrak, Hawk, and UpLink are trademarks of RadiSys Corporation.

† All other trademarks, registered trademarks, service marks, and trade names are the property of their respective owners.

Contents

Low-Level System Configuration Module (cnfgdata) 15	
Overview	16
cnfgdata Module Field Configuration Options	16
Direct Modification Advantages	16
Description File/Rebuild Advantages	16
Direct Modification	17
Description File Modification	19
Low-Level Configuration Module Field Reference	19
Module Header Fields	20
_m_group	22
_m_user	23
mod_name	24
m_access	25
m_tylan	27
m_attrev	29
m_edit	31
Console Device Fields	31
console_name	33
cons_vector	34
cons_priority	35
cons_level	36
cons_timeout	37
cons_parity	38
cons_baudrate	39
cons_wordsize	41
cons_stopbits	42
cons_flow	43
Communication Device Fields	43
comm_name	45
cons_vector	46
cons_priority	47
cons_level	48
cons_timeout	49
cons_parity	50
cons_baudrate	51
cons_wordsize	53
cons_stopbits	54
cons_flow	55
Debugger Fields	55
debug_name	57
debug_call_at_cold	58
Low-Level Protocol Manager Fields	58

maxllpmprotos	60
maxrcvmbufs	61
maxllpmconns	62
llpm_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
Configuration Boot Data Fields.....	76
boot_count	78
boot_cmdsize	79
Boot Data Fields.....	79
boot_abname	80
boot_newab	81
boot_newname	82
boot_automenu	83
boot_params	84
autoboot_delay	85
Notification Services Field	85
max_notifiers	86
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
Init Module Field Reference	92
Module Header Fields.....	93
_m_group	94
_m_user	95
mod_name	96
m_access	97
m_tylan	99
m_attrev	101
m_edit	103
Module Body Fields.....	103
m_site	106
m_cputyp	107
install_name	108
os9rev_name	109



sysgo_name	110
sparam_string	111
drive_name	112
console_name	113
extens_list	114
ticker_name	115
rtc_name	116
ioman_name	117
acct_name	118
m_procs	119
m_paths	120
m_events	121
m_ticksec	122
m_slice	123
m_syspri	124
m_minpty	125
m_maxage	126
m_dsptbl	127
m_cpucompat	128
m_tmzone	129
m_level	130
m_major	131
m_minor	132
m_edition	133
m_compat	134
m_maxsigs	135
preio_name	136
Memlist Fields.....	136
type	138
prior	139
access	140
blksiz	141
lolim	142
hilim	143
desc	144
dma_addr	145
Cachelist Fields.....	145
blk_beg	146
blk_end	147
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

mod_name	158
m_access	159
m_tylan	161
m_attrev	163
m_edit	165
Device Descriptor Data Definition Fields.....	165
dd_port	167
dd_lun	168
dd_pd_size	169
dd_type	170
dd_mode	172
fmgr_name	174
drv_name	175
dd_class	176
SCF Description Block Fields.....	176
outdev_name	178
SCF Logical Unit Static Storage Fields.....	178
hardware_vector	180
v_irqlevel	181
v_priority	182
v_pollin	183
v_pollout	184
v_lun	185
v_irqmask	186
v_maxbuff	187
v_insize	188
v_outsize	189
v_line	190
v_intr	191
v_quit	194
v_psch	195
v_xon	196
v_xoff	197
v_baud	198
v_parity	200
v_stopbits	201
v_wordsize	202
v_rtsstate	203
v_devspec	204
SCF Path Option Fields.....	204
pd_inmap0type	211
pd_inmap0func_code	212
pd_inmap0size	214
pd_inmap0string	215
pd_inmap1type	216
pd_inmap1func_code	217
pd_inmap1size	218
pd_inmap1string	219
pd_inmap2type	220



pd_inmap2func_code	221
pd_inmap2size	222
pd_inmap2string	223
pd_inmap3type	224
pd_inmap3func_code	225
pd_inmap3size	226
pd_inmap3string	227
pd_inmap4type	228
pd_inmap4func_code	229
pd_inmap4size	230
pd_inmap4string	231
pd_inmap5type	232
pd_inmap5func_code	233
pd_inmap5size	234
pd_inmap5string	235
pd_inmap6type	236
pd_inmap6func_code	237
pd_inmap6size	238
pd_inmap6string	239
pd_inmap7type	240
pd_inmap7func_code	241
pd_inmap7size	242
pd_inmap7string	243
pd_inmap8type	244
pd_inmap8func_code	245
pd_inmap8size	246
pd_inmap8string	247
pd_inmap9type	248
pd_inmap9func_code	249
pd_inmap9size	250
pd_inmap9string	251
pd_inmap10type	252
pd_inmap10func_code	253
pd_inmap10size	254
pd_inmap10string	255
pd_inmap11type	256
pd_inmap11func_code	257
pd_inmap11size	258
pd_inmap11string	259
pd_inmap12type	260
pd_inmap12func_code	261
pd_inmap12size	262
pd_inmap12string	263
pd_inmap13type	264
pd_inmap13func_code	265
pd_inmap13size	266
pd_inmap13string	267
pd_inmap14type	268
pd_inmap14func_code	269

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_inma22func_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



pd_inmap26string	319
pd_inmap27type	320
pd_inmap27func_code	321
pd_inmap27size	322
pd_inmap27string	323
pd_inmap28type	324
pd_inmap28func_code	325
pd_inmap28size	326
pd_inmap28string	327
pd_inmap29type	328
pd_inmap29func_code	329
pd_inmap29size	330
pd_inmap29string	331
pd_inmap30type	332
pd_inmap30func_code	333
pd_inmap30size	334
pd_inmap30string	335
pd_inmap31type	336
pd_inmap31func_code	337
pd_inmap31size	338
pd_inmap31string	339
pd_eorch	340
pd_eofch	343
pd_tabch	344
pd_bellch	345
pd_bspch	346
pd_case	347
pd_backsp	348
pd_delete	349
pd_echo	350
pd_alf	351
pd_pause	352
pd_insm	353
pd_nulls	354
pd_page	355
pd_tabsiz	356

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_blksiz	385
pd_flags	386
pd_dmamode	387
pd_sci_id	388
pd_scsilun	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



pd_format	425
pd_cyl	427
pd_blk	428
pd_tob	429
pd_sas	430
pd_ilv	431
pd_toffs	432
pd_boffs	433
pd_trys	434
pd_bsize	435
pd_cntl	436
pd_wpc	437
pd_rwr	438
pd_park	439
pd_lsnofts	440
pd_xfersize	441
RBF Logical Unit Static Storage Fields	441
v_vector	442
v_irqlevel	443
v_priority	444
RBF Logical Unit Options	444
lu_stp	445
lu_tfm	446
lu_lun	447
lu_ctrlrid	448
lu_totcyls	449
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
dvr_name	476

dd_class	477
PCF Path Option Fields.....	477
pd_sid	479
pd_vfy	480
pd_format	481
pd_cyl	483
pd_blk	484
pd_tob	485
pd_sas	486
pd_ilv	487
pd_toffs	488
pd_boffs	489
pd_trys	490
pd_bsize	491
pd_cntl	492
pd_wpc	493
pd_rwr	494
pd_park	495
pd_lsnooffs	496
pd_xfersize	497
PCF Logical Unit Static Storage Fields	497
v_vector	498
v_irqlevel	499
v_priority	500
PCF Logical Unit Options	500
lu_stp	501
lu_tfm	502
lu_lun	503
lu_ctrlrid	504
lu_totcyls	505
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_attrev	520
m_edit	522
Device Descriptor Data Definition Fields.....	522
dd_port	524
dd_lun	525
dd_pd_size	526




dd_type	527
dd_mode	529
fmgr_name	531
drv_name	532
dd_class	533
Pipeman Logical Unit Static Storage.....	533
bufsz	535

Index



1

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

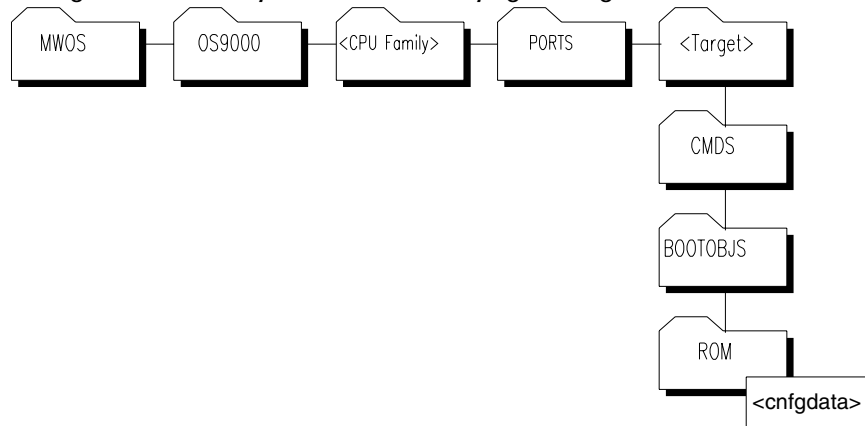
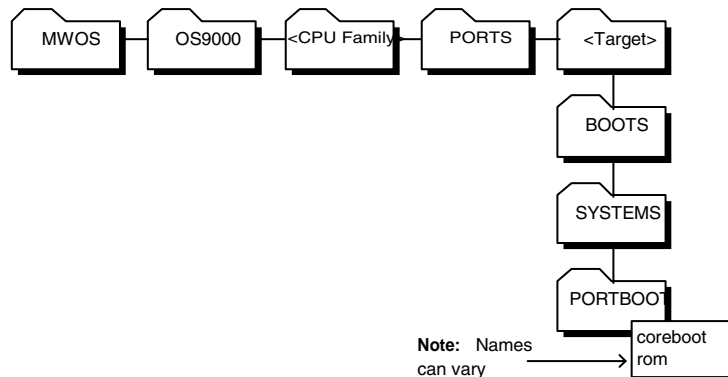


Figure 1-2. Directory Location for Modifying the `cnfgdata` Module as Part of a Boot Image



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/BOOTBJS/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)
.
```

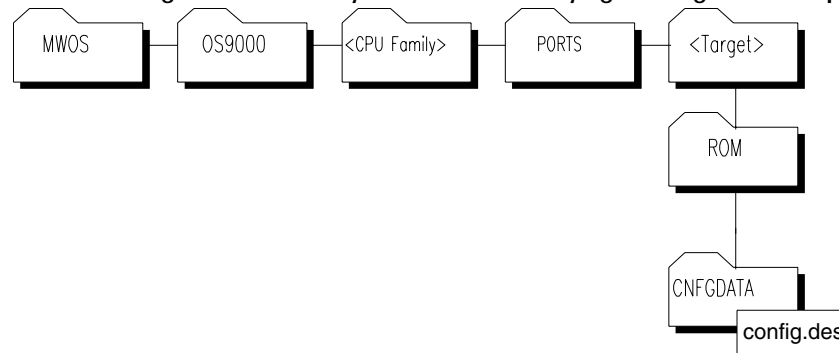
```
Which? [?/1-19/p/t/a/w/q] w
```

```
Which? [?/1-19/p/t/a/w/q] q
```

Description File Modification

You can use these procedures to modify the `cnfgdata` description file and rebuild the `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



Description File Modification Procedures

1. Change to the ROM/CNFGDATA directory (see [Figure 1-3](#)).
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 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 `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 ROM/CNFGDATA directory:

```
os9make
```

7. Rebuild your boot image to include the new `cnfgdata` module.

Low-Level Configuration Module Field Reference

This section contains a list of all configurable fields in the `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 1-1. Module Header Fields

Field	Description File Macro
<code>_m_group</code>	<code>MH_GROUP</code>
<code>_m_user</code>	<code>MH_USER</code>
<code>mod_name</code>	<code>MH_NAME</code>
<code>m_access</code>	<code>MH_ACCESS</code>
<code>m_tylan</code>	<code>MH_TYLAN</code>

Table 1-1. Module Header Fields (Continued)

Field	Description File Macro
<code>m_attrev</code>	MH_ATTREV
<code>m_edit</code>	MH_EDIT

_m_group
MH_GROUP

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

_m_user
MH_USER

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

mod_name
MH_NAME

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).

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 1-2. m_access Available Values

Description	Macro	EditMod
Read permission by owner	MP_OWNER_READ	0x0001
Write permission by owner	MP_OWNER_WRITE	0x0002
Execute permission by owner	MP_OWNER_EXEC	0x0004
Owner permission mask	MP_OWNER_MASK	0x000f
Read permission by group	MP_GROUP_READ	0x0010
Write permission by group	MP_GROUP_WRITE	0x0020
Execute permission by group	MP_GROUP_EXEC	0x0040
Group permission mask	MP_GROUP_MASK	0x00f0

Table 1-2. m_access Available Values (Continued)

Description	Macro	EditMod
Read permission by world	MP_WORLD_READ	0x0100
Write permission by world	MP_WORLD_WRITE	0x0200
Execute permission by world	MP_WORLD_EXEC	0x0400
World permission mask	MP_WORLD_MASK	0x0f00
All permissions for owner, group, and world	MP_WORLD_ACCESS	0x0777
System permission mask	MP_SYSTM_MASK	0xf000

m_tylan
MH_TYLAN

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 1-3. `m_tylan` Available Module Type Values

Description	Macro	EditMod
Not used (wildcard value in system calls)	MT_ANY	0x0000
Program module	MT_PROGRAM	0x0001
Subroutine module	MT_SUBROUT	0x0002
Multi-module (reserved for future use)	MT_MULTI	0x0003
Data module	MT_DATA	0x0004
Configuration data block data module	MT_CDBDATA	0x0005
Reserved for future use	0xb-0xa	0xb-0xa
User trap library	MT_TRAPLIB	0x000b

Table 1-3. `m_tylan` Available Module Type Values (Continued)

Description	Macro	EditMod
System module	MT_SYSTEM	0x000c
File manager module	MT_FILEMAN	0x000d
Physical device driver	MT_DEVDRVR	0x000e
Device descriptor module	MT_DEVDESC	0x000f
User definable	0x10-0xfe	0x10-0xfe
Module type mask	MT_MASK	0xff00

Table 1-4. `m_tylan` Available Language Code Values

Description	Macro	EditMod
Unspecified language (wildcard in system calls)	ML_ANY	0x0
Machine language	ML_OBJECT	0x1
Basic I-code (reserved for future use)	ML_ICODE	0x2
Pascal P-code (reserved for future use)	ML_PCODE	0x3
C I-code (reserved for future use)	ML_CCODE	0x4
Cobol I-code (reserved for future use)	ML_CBLCODE	0x5
Fortran	ML_FRTNCODE	0x6
Reserved for future use	0x7-0xf	0x7-0xf
User-definable	0x10-0xfe	0x10-0xfe
Module language mask	ML_MASK	0x00ff

m_attrev
MH_ATTREV

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 1-5. m_attrev Available Attribute and Revision Values

Description	Macro	EditMod
The module is re-entrant (sharable by multiple tasks).	MA_REENT (shifted left to first byte: MA_REENT<<8)	0x80 (shifted left to first byte: 0x8000)
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.	MA_GHOST (shifted left to first byte: MA_GHOST<<8)	0x40 (shifted left to first byte: 0x4000)

Table 1-5. `m_attrrev` Available Attribute and Revision Values

Description	Macro	EditMod
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

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

Field	Description File Macro
<code>console_name</code>	CONS_NAME
<code>cons_vector</code>	CONS_VECTOR
<code>cons_priority</code>	CONS_PRIORITY
<code>cons_level</code>	CONS_LEVEL
<code>cons_timeout</code>	CONS_TIMEOUT

Table 1-6. Console Device Fields (Continued)

Field	Description File Macro
<code>cons_parity</code>	CONS_PARITY
<code>cons_baudrate</code>	CONS_BAUDRATE
<code>cons_wordsize</code>	CONS_WORDSIZE
<code>cons_stopbits</code>	CONS_STOPBITS
<code>cons_flow</code>	CONS_FLOW

console_name
CONS_NAME

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/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
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

cons_priority
CONS_PRIORITY

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 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**.

cons_level
CONS_LEVEL

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
CONS_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

cons_parity
CONS_PARITY

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 1-7.

Table 1-7. **cons_parity** Available Values

Description	Macro	EditMod
No parity	CONS_NOPARITY	0x00
Odd parity	CONS_ODDPARITY	0x01
Even parity	CONS_EVENPARITY	0x02
Mark parity	CONS_MARKPARITY	0x03
Space parity	CONS_SPACEPARITY	0x04
Parity mask	CONS_PARITY_MASK	0x0F
Parity shift	CONS_PARITY_SHIFT	0

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

Description	Macro	EditMod
Hardwire baud rate	CONS_BAUDRATE_HARDWIRE	0x00
50 bits per second (bps)	CONS_BAUDRATE_50	0x01
75 bps	CONS_BAUDRATE_75	0x02
110 bps	CONS_BAUDRATE_110	0x03
134.5 bps	CONS_BAUDRATE_134P5	0x04
150 bps	CONS_BAUDRATE_150	0x05
300 bps	CONS_BAUDRATE_300	0x06
600 pbs	CONS_BAUDRATE_600	0x07
1200 bps	CONS_BAUDRATE_1200	0x08
1800 bps	CONS_BAUDRATE_1800	0x09

Table 1-8. `cons_baudrate` Available Values (Continued)

Description	Macro	EditMod
2000 bps	CONS_BAUDRATE_2000	0x0A
2400 bps	CONS_BAUDRATE_2400	0x0B
3600 bps	CONS_BAUDRATE_3600	0x0C
4800 bps	CONS_BAUDRATE_4800	0x0D
7200 bps	CONS_BAUDRATE_7200	0x0E
9600 bps	CONS_BAUDRATE_9600	0x0F
19,200 bps	CONS_BAUDRATE_19200	0x10
31,250 bps	CONS_BAUDRATE_31250	0x11
38,400 bps	CONS_BAUDRATE_38400	0x12
56,000 bps	CONS_BAUDRATE_56000	0x13
57,600 bps	CONS_BAUDRATE_57600	0x14
64,000 bps	CONS_BAUDRATE_64000	0x15
115,200 bps	CONS_BAUDRATE_115200	0x16
No echo	CONS_NOECHO	0x80
Baud rate mask	CONS_BAUDRATE_MASK	0x3F

cons_wordsize
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/CNFGDATA/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 1-9. **cons_wordsize** Available Values

Description	Macro	EditMod
8 bit word size	CONS_8BITS	0x00
7 bit word size	CONS_7BITS	0x40
6 bit word size	CONS_6BITS	0x80
5 bit word size	CONS_5BITS	0xC0
Word size mask	CONS_DBITS_MASK	0xC0
Word size shift	CONS_DBITS_SHIFT	6

cons_stopbits
CONS_STOPBITS

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 1-10. **cons_stopbits** Available Values

Description	Macro	EditMod
Stop bit length of 1	CONS_1STOP	0x00
Stop bit length of 1.5	CONS_1P5STOP	0x10
Stop bit length of 2	CONS_2STOP	0x20
Stop bit mask	CONS_STOP_MASK	0x30
Stop bit shift	CONS_STOP_SHIFT	0x40
Stop data bit shift	CONS_DBITS_SHIFT	0x60

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

Table 1-11. **cons_flow** Available Values

Description	Macro	EditMod
No handshaking	CONS_NOSHAKE	0x00
XOFF, any character on	CONS_SWSHAKE	0x01
Hardware handshaking	CONS_HWSHAKE	0x02
Strictly XON-XOFF	CONS_SWSTRICT	0x03

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 1-12. Communication Device Fields

Field	Description File Macro
<code>comm_name</code>	COMM_NAME
<code>cons_vector</code>	COMM_VECTOR
<code>cons_priority</code>	COMM_PRIORITY
<code>cons_level</code>	COMM_LEVEL
<code>cons_timeout</code>	COMM_TIMEOUT
<code>cons_parity</code>	COMM_PARITY
<code>cons_baudrate</code>	COMM_BAUDRATE
<code>cons_wordsize</code>	COMM_WORDSIZE
<code>cons_stopbits</code>	COMM_STOPBITS
<code>cons_flow</code>	COMM_FLOW

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
COMM_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

cons_priority
COMM_PRIORITY

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**.

cons_level
COMM_LEVEL

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

cons_parity
COMM_PARITY

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 1-13. **cons_parity** Available Values

Description	Macro	EditMod
No parity	CONS_NOPARITY	0x00
Odd parity	CONS_ODDPARITY	0x01
Even parity	CONS_EVENPARITY	0x02
Mark parity	CONS_MARKPARITY	0x03
Space parity	CONS_SPACEPARITY	0x04
Parity mask	CONS_PARITY_MASK	0x0F
Parity shift	CONS_PARITY_SHIFT	0

cons_baudrate COMM_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 1-14.

Table 1-14. cons_baudrate Available Values

Description	Macro	EditMod
Hardwire baud rate	CONS_BAUDRATE_HARDWIRE	0x00
50 bits per second (bps)	CONS_BAUDRATE_50	0x01
75 bps	CONS_BAUDRATE_75	0x02
110 bps	CONS_BAUDRATE_110	0x03
134.5 bps	CONS_BAUDRATE_134P5	0x04
150 bps	CONS_BAUDRATE_150	0x05
300 bps	CONS_BAUDRATE_300	0x06
600 pbs	CONS_BAUDRATE_600	0x07
1200 bps	CONS_BAUDRATE_1200	0x08
1800 bps	CONS_BAUDRATE_1800	0x09

Table 1-14. `cons_baudrate` Available Values (Continued)

Description	Macro	EditMod
2000 bps	CONS_BAUDRATE_2000	0x0A
2400 bps	CONS_BAUDRATE_2400	0x0B
3600 bps	CONS_BAUDRATE_3600	0x0C
4800 bps	CONS_BAUDRATE_4800	0x0D
7200 bps	CONS_BAUDRATE_7200	0x0E
9600 bps	CONS_BAUDRATE_9600	0x0F
19,200 bps	CONS_BAUDRATE_19200	0x10
31,250 bps	CONS_BAUDRATE_31250	0x11
38,400 bps	CONS_BAUDRATE_38400	0x12
56,000 bps	CONS_BAUDRATE_56000	0x13
57,600 bps	CONS_BAUDRATE_57600	0x14
64,000 bps	CONS_BAUDRATE_64000	0x15
115,200 bps	CONS_BAUDRATE_115200	0x16
No echo	CONS_NOECHO	0x80
Baud rate mask	CONS_BAUDRATE_MASK	0x3F

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 1-15. **cons_wordsize** Available Values

Description	Macro	EditMod
8 bit word size	CONS_8BITS	0x00
7 bit word size	CONS_7BITS	0x40
6 bit word size	CONS_6BITS	0x80
5 bit word size	CONS_5BITS	0xC0
Word size mask	CONS_DBITS_MASK	0xC0
Word size shift	CONS_DBITS_SHIFT	6

cons_stopbits
COMM_STOPBITS

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 1-16. **cons_stopbits** Available Values

Description	Macro	EditMod
Stop bit length of 1	CONS_1STOP	0x00
Stop bit length of 1.5	CONS_1P5STOP	0x10
Stop bit length of 2	CONS_2STOP	0x20
Stop bit mask	CONS_STOP_MASK	0x30
Stop bit shift	CONS_STOP_SHIFT	0x40
Stop data bit shift	CONS_DBITS_SHIFT	0x60

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 1-17. cons_flow Available Values

Description	Macro	EditMod
No handshaking	CONS_NOSHAKE	0x00
XOFF, any character on	CONS_SWSHAKE	0x01
Hardware handshaking	CONS_HWSHAKE	0x02
Strictly XON-XOFF	CONS_SWSTRICT	0x03

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 1-18. Debugger Fields

Field	Description File Macro
<code>debug_name</code>	DEBUGGER_NAME
<code>debug_call_at_cold</code>	DEBUGGER_COLD_FLAG

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 1-19. `debug_call_at_cold` Available Values

Description	Macro	EditMod
Bypass calling debugger during boot sequence	DEBUG_BYPASS	0x0
Call debugger during boot sequence	DEBUG_CALL	0x1

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

Field	Description File Macro
<code>maxllpmprotos</code>	LLPM_MAXPROTOS
<code>maxrcvmbufs</code>	LLPM_MAXRCVMBUFS
<code>maxllpmconns</code>	LLPM_MAXCONNS
<code>llpm_count</code>	LLPM_COUNT

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

maxrcvmbufs
LLPM_MAXRCVMBUFS**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

llpm_count
LLPM_COUNT

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 1-21. Interface Data Fields

Field	Description
<code>ip_address</code>	Low-level IP address
<code>subnet_mask</code>	Low-level subnet mask
<code>brdcst_address</code>	Low-level broadcast address
<code>gw_address</code>	Low-level gateway address
<code>mac_address</code>	Low-level MAC address
<code>hwtype</code>	Low-level interface data driver type

Table 1-21. Interface Data Fields (Continued)

Field	Description
<code>if_flags</code>	Interface flags
<code>if_name</code>	Low-level protocol manager name
<code>port_address</code>	Low-level protocol manager physical address
<code>if_vector</code>	Low-level protocol manager vector number
<code>if_priority</code>	Low-level protocol manager polling priority
<code>if_level</code>	Low-level protocol manager hardware priority

ip_address

Low-level IP 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>]
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 1-22. hwtype Available Values

Description	Macro	EditMod
No driver type	LLPM_NOHW	0x0
SLIP driver type	LLPM_SLIP	0x1
Ethernet driver type	LLPM_ETHER	0x2

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 1-23. `if_flags` Available Values

Description	Macro	EditMod
Applies only to SLIP array entries.	LLIF_CSLIP_ON	0x8000
Applies only to SLIP array entries.	LLIF_CSLIP_OFF	0x0000

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 ValueRefer 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 1-24. Configuration Boot Data Fields

Field	Description File Macro
<code>boot_count</code>	BOOT_COUNT
<code>boot_cmdsize</code>	BOOT_CMDSIZE

boot_count
BOOT_COUNT

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

boot_cmdsize
BOOT_CMDSIZE

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

Field	Description
<code>boot_abname</code>	Abbreviated booter name
<code>boot_newab</code>	New abbreviated booter name
<code>boot_newname</code>	Optional replacement full name
<code>boot_automenu</code>	Booter types for registration
<code>boot_params</code>	Optional parameter string
<code>autoboot_delay</code>	Autoboot delay value

boot_abname Abbreviated Booter Name

EditMod Labels

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

Description

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_newab

New Abbreviated Booter Name

EditMod Labels

2-configuration data
5-boot services data structure
2-boot data
<n>-boot data [<n>]
2-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).

boot_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 1-26. `boot_automenu` Available Values

Description	config.des Macro	EditMod Hex
Auto booter	BT_AUTO	0x1
Menu booter	BT_MENU	0x2

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

Field	Description File Macro
<code>max_notifiers</code>	<code>MAX_NOTIFIERS</code>

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**.

2

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.

Figure 2-1. Directory Location for Modifying the init Module as a Stand-alone Module

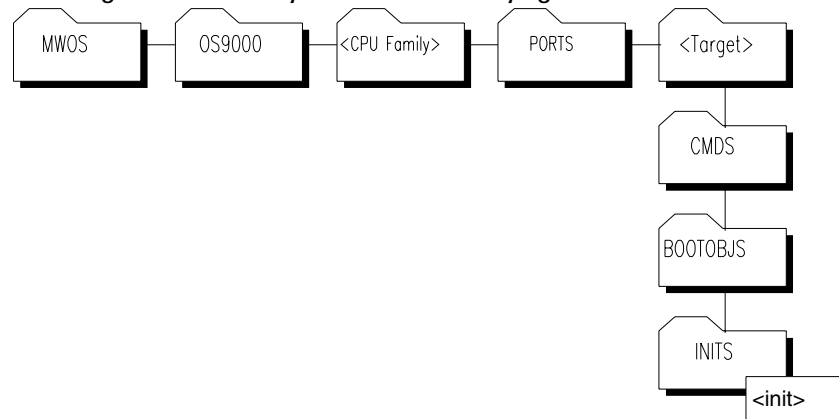
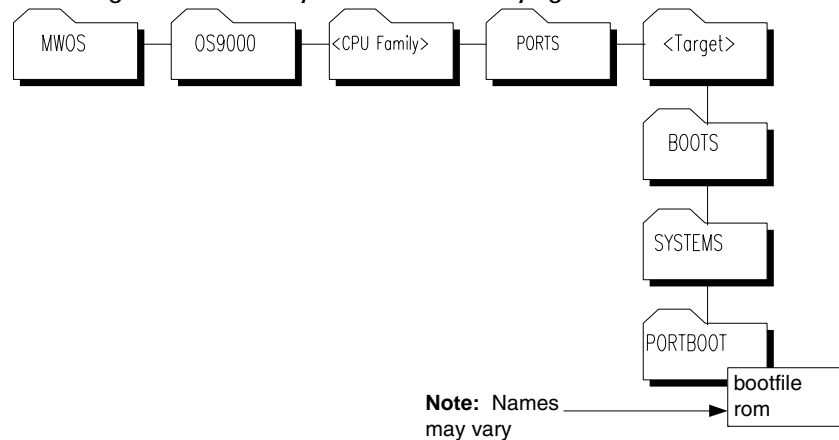


Figure 2-2. Directory Location for Modifying the init Module as Part of a Boot Image



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/BOOTOBS/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

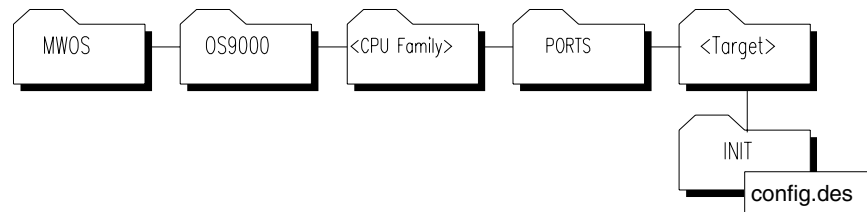
1. installation site code	: 0x0
2. cpu class	: 0x1bc7
3. installation string	: "PS7111"
4. OS-9000 level/revision string	: "OS-9000 for the ARM"
5. initial module name	: "shell"
6. parameter list	: ""
7. system RBF device	: ""
8. system SCF device	: "/term"
9. customization module list	: "OS9P2 fpu ssm"
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	

```
$Which? [?/1-6/p/t/a/w/q]
.
.
.
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 `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 `sysgo` 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 begins with a standard module header. The module header's `m_exec` offset is a pointer to the system's constant table. The `init` fields are defined in the `init.h` header file.



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 2-1. Module Header Fields

Field	Description File Macro
<code>_m_group</code>	<code>MH_GROUP</code>
<code>_m_user</code>	<code>MH_USER</code>
<code>mod_name</code>	<code>MH_NAME</code>
<code>m_access</code>	<code>MH_ACCESS</code>
<code>m_tylan</code>	<code>MH_TYLAN</code>
<code>m_attrev</code>	<code>MH_ATTREV</code>
<code>m_edit</code>	<code>MH_EDITION</code>

_m_group
MH_GROUP

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

`_m_user`
`MH_USER`

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

mod_name
MH_NAME

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 2-2. m_access Available Values

Description	Macro	EditMod
Read permission by owner	MP_OWNER_READ	0x0001
Write permission by owner	MP_OWNER_WRITE	0x0002
Execute permission by owner	MP_OWNER_EXEC	0x0004
Owner permission mask	MP_OWNER_MASK	0x000f
Read permission by group	MP_GROUP_READ	0x0010
Write permission by group	MP_GROUP_WRITE	0x0020
Execute permission by group	MP_GROUP_EXEC	0x0040

Table 2-2. `m_access` Available Values (Continued)

Description	Macro	EditMod
Group permission mask	MP_GROUP_MASK	0x00f0
Read permission by world	MP_WORLD_READ	0x0100
Write permission by world	MP_WORLD_WRITE	0x0200
Execute permission by world	MP_WORLD_EXEC	0x0400
World permission mask	MP_WORLD_MASK	0x0f00
All permissions for owner, group, and world	MP_WORLD_ACCESS	0x0777
System permission mask	MP_SYSTM_MASK	0xf000

m_tylan
MH_TYLAN

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 2-3. m_tylan Available Module Type Values

Description	Macro	EditMod
Not used (wildcard value in system calls)	MT_ANY	0x0000
Program module	MT_PROGRAM	0x0001
Subroutine module	MT_SUBROUT	0x0002
Multi-module (reserved for future use)	MT_MULTI	0x0003
Data module	MT_DATA	0x0004
Configuration data block data module	MT_CDBDATA	0x0005
Reserved for future use	0xb-0xa	0xb-0xa

Table 2-3. `m_tylan` Available Module Type Values (Continued)

Description	Macro	EditMod
User trap library	MT_TRAPLIB	0x000b
System module	MT_SYSTEM	0x000c
File manager module	MT_FILEMAN	0x000d
Physical device driver	MT_DEVDRVR	0x000e
Device descriptor module	MT_DEVDESC	0x000f
User definable	0x10-0xfe	0x10-0xfe
Module type mask	MT_MASK	0xff00

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

Description	Macro	EditMod
Unspecified language (wildcard in system calls)	ML_ANY	0x0
Machine language	ML_OBJECT	0x1
Basic I-code (reserved for future use)	ML_ICODE	0x2
Pascal P-code (reserved for future use)	ML_PCODE	0x3
C I-code (reserved for future use)	ML_CCODE	0x4
Cobol I-code (reserved for future use)	ML_CBLCODE	0x5
Fortran	ML_FRTNCODE	0x6
Reserved for future use	0x7-0xf	0x7-0xf
User-definable	0x10-0xfe	0x10-0xfe
Module language mask	ML_MASK	0x00ff

m_attrrev
MH_ATTREV

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_attrrev Available Attribute and Revision Values

Description	Macro	EditMod
The module is re-entrant (sharable by multiple tasks).	MA_REENT (shifted left to first byte: MA_REENT<<8)	0x80 (shifted left to first byte: 0x8000)
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.	MA_GHOST (shifted left to first byte: MA_GHOST<<8)	0x40 (shifted left to first byte: 0x4000)

Table 2-5. `m_attrrev` Available Attribute and Revision Values (Continued)

Description	Macro	EditMod
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

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. 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 2-6. Module Header Fields

Field	Description File Macro
m_site	SITE
m_cputyp	MPUCHIP
install_name	INSTALNAME
os9rev_name	OS9K_REVSTR
sysgo_name	SYS_START
sparam_string	SYS_PARAMS

Table 2-6. Module Header Fields (Continued)

Field	Description File Macro
<code>drive_name</code>	<code>SYS_DEVICE</code>
<code>console_name</code>	<code>CONS_NAME</code>
<code>extens_list</code>	<code>EXTENSIONS</code>
<code>ticker_name</code>	<code>TICK_NAME</code>
<code>rtc_name</code>	<code>RTC_NAME</code>
<code>ioman_name</code>	<code>IOMAN_NAME</code>
<code>acct_name</code>	<code>USRACCT_NAME</code>
<code>m_procs</code>	<code>PROCS</code>
<code>m_paths</code>	<code>PATHS</code>
<code>m_events</code>	<code>EVENTS</code>
<code>m_ticksec</code>	<code>TICK_SEC</code>
<code>m_slice</code>	<code>SLICE</code>
<code>m_syspri</code>	<code>SYS_PRIOR</code>
<code>m_minpty</code>	<code>MINPTY</code>
<code>m_maxage</code>	<code>MAXPTY</code>
<code>m_dsptbl</code>	<code>DSPTBLSZ</code>
<code>m_cpucompat</code>	<code>CPUCOMPAT</code>
<code>m_tmzone</code>	<code>SYS_TMZONE</code>
<code>m_level</code>	<code>OS_LEVEL</code>
<code>m_major</code>	<code>OS_VERSION</code>
<code>m_minor</code>	<code>OS_REVISION</code>
<code>m_edition</code>	<code>OS_EDITION</code>

Table 2-6. Module Header Fields (Continued)

Field	Description File Macro
<code>m_compat</code>	COMPAT
<code>m_maxsigs</code>	MAXSIGS
<code>preio_name</code>	PREIOS_NAME

m_site
SITE**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


m_cputyp
MPUCHIP**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`

install_name
INSTALNAME

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`).

os9rev_name
OS9K_REVSTR

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`).

sysgo_name
SYS_START

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`).

drive_name
SYS_DEVICE

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`).

console_name
CONS_NAME

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).

ticker_name
TICK_NAME

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`).

rtc_name
RTC_NAME

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`).

ioman_name
IOMAN_NAME

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`

m_paths
PATHS**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`

m_events
EVENTS**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

m_ticksec
TICK_SEC

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

m_slice
SLICE**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

m_syspri
SYS_PRIOR

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

m_minpty
MINPTY

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

m_dsptbl
DSPTBLSZ**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`

m_cpucompat
CPUCOMPAT

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

m_level
OS_LEVEL

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

m_major
OS_VERSION

EditMod Labels

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

m_minor
OS_REVISION

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

m_edition
OS_EDITION

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 2-7. m_compat Available Compatibility Flags

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

m_maxsig
MAXSIG

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

preio_name
PREIOS_NAME

EditMod Labels

2-init module contents

32-pre-IO 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 2-8. Memlist Fields

Field	Description
<code>type</code>	Memory type code
<code>prior</code>	Memory allocation priority
<code>access</code>	Access permissions
<code>blksiz</code>	Search block size
<code>lolim</code>	Beginning block address

Table 2-8. Memlist Fields (Continued)

Field	Description
<code>hilim</code>	Ending block address
<code>desc</code>	Memory list description
<code>dma_addr</code>	External bus address

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 2-9. type Available Values for Memory Lists

Description	Macro	EditMod
System RAM memory	MEM_SYS	0x01
Shared memory (0x8000 - 0xffff)	MEM_SHARED	0x8000



prior Memory Allocation Priority

EditMod Labels

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

access Access Permissions

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](#).



Only `B_USERRAM` memory can be initialized.

Table 2-10. Access Bit Definitions for Memory Type

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

blksiz

Search Block Size

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

lolim

Beginning Block Address

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

hilim

Ending Block Address

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`).

dma_addr
External Bus Address

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 2-11. Cachelist Fields

Field	Description
<code>blk_beg</code>	Beginning address of memory region
<code>blk_end</code>	Ending address of memory region

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



3

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.

Figure 3-1. Directory Location for Modifying an SCF Device Descriptor

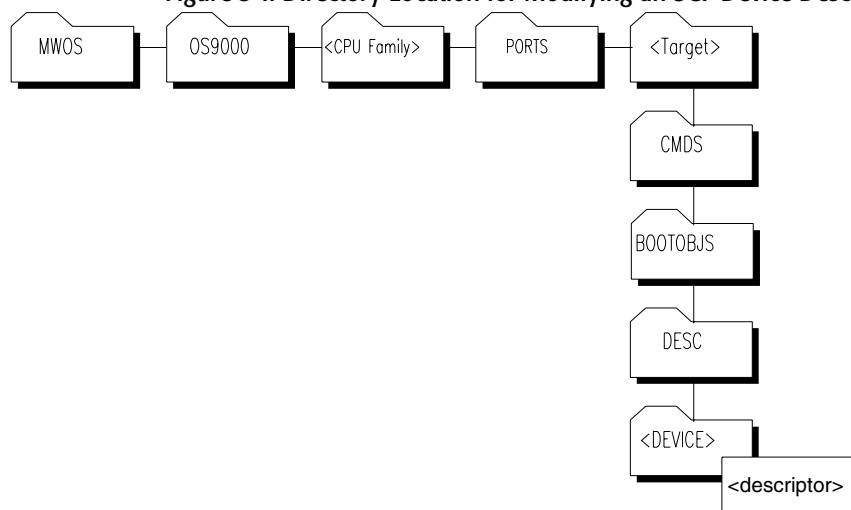
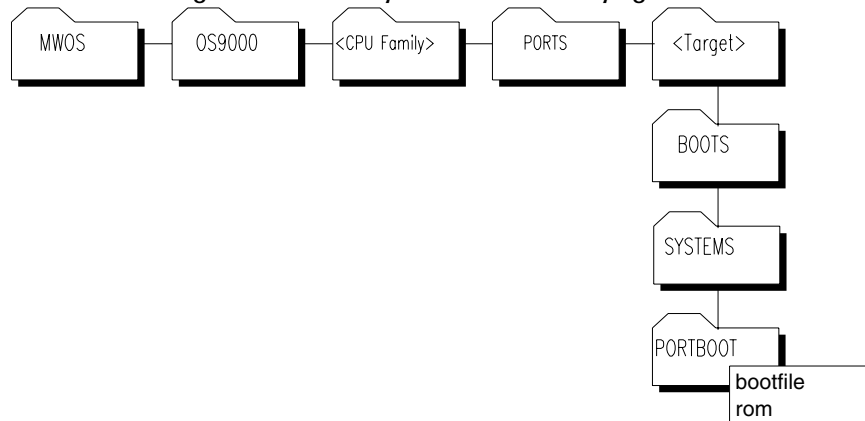


Figure 3-2. Directory Location for Modifying Low-Level Boot Images



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. Go to the CMDS/BOOTOBS/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/SYSTEMS/PORTBOOT directory (see [Figure 3-2](#)).
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 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
```



```

irq polling priority           : 5
New value: 8

irq polling priority           : 8

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

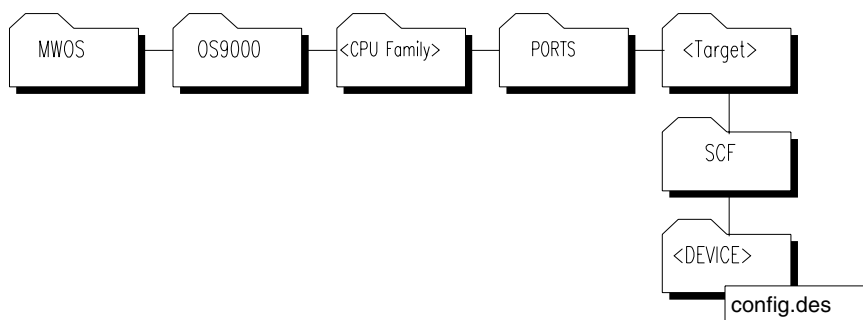
Which? [?/1-19/p/t/a/w/q] q

```

Description File Modification

You can use these procedures to modify the 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:

```
os9make
```

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 3-1. Module Header Fields

Field	Description File Macro
<code>_m_group</code>	<code>MH_GROUP</code>
<code>_m_user</code>	<code>MH_USER</code>

Table 3-1. Module Header Fields (Continued)

Field	Description File Macro
<code>mod_name</code>	<code>MH_NAME</code>
<code>m_access</code>	<code>MH_ACCESS</code>
<code>m_tylan</code>	<code>MH_TYLAN</code>
<code>m_attrev</code>	<code>MH_ATTREV</code>
<code>m_edit</code>	<code>MH_EDITION</code>

_m_group
MH_GROUP

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

_m_user
MH_USER

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

mod_name
MH_NAME

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).

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 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 3-2. m_access Available Values

Description	Macro	EditMod
Read permission by owner	MP_OWNER_READ	0x0001
Write permission by owner	MP_OWNER_WRITE	0x0002
Execute permission by owner	MP_OWNER_EXEC	0x0004
Owner permission mask	MP_OWNER_MASK	0x000f
Read permission by group	MP_GROUP_READ	0x0010
Write permission by group	MP_GROUP_WRITE	0x0020
Execute permission by group	MP_GROUP_EXEC	0x0040

Table 3-2. `m_access` Available Values (Continued)

Description	Macro	EditMod
Group permission mask	MP_GROUP_MASK	0x00f0
Read permission by world	MP_WORLD_READ	0x0100
Write permission by world	MP_WORLD_WRITE	0x0200
Execute permission by world	MP_WORLD_EXEC	0x0400
World permission mask	MP_WORLD_MASK	0x0f00
All permissions for owner, group, and world	MP_WORLD_ACCESS	0x0777
System permission mask	MP_SYSTM_MASK	0xf000

m_tylan
MH_TYLAN

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 3-3. m_tylan Available Module Type Values

Description	Macro	EditMod
Not used (wildcard value in system calls)	MT_ANY	0x0000
Program module	MT_PROGRAM	0x0001
Subroutine module	MT_SUBROUT	0x0002
Multi-module (reserved for future use)	MT_MULTI	0x0003
Data module	MT_DATA	0x0004
Configuration data block data module	MT_CDBDATA	0x0005
Reserved for future use	0xb-0xa	0xb-0xa

Table 3-3. `m_tylan` Available Module Type Values (Continued)

Description	Macro	EditMod
User trap library	MT_TRAPLIB	0x000b
System module	MT_SYSTEM	0x000c
File manager module	MT_FILEMAN	0x000d
Physical device driver	MT_DEVDRVR	0x000e
Device descriptor module	MT_DEVDESC	0x000f
User definable	0x10-0xfe	0x10-0xfe
Module type mask	MT_MASK	0xff00

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

Description	Macro	EditMod
Unspecified language (wildcard in system calls)	ML_ANY	0x0
Machine language	ML_OBJECT	0x1
Basic I-code (reserved for future use)	ML_ICODE	0x2
Pascal P-code (reserved for future use)	ML_PCODE	0x3
C I-code (reserved for future use)	ML_CCODE	0x4
Cobol I-code (reserved for future use)	ML_CBLCODE	0x5
Fortran	ML_FRTNCODE	0x6
Reserved for future use	0x7-0xf	0x7-0xf
User-definable	0x10-0xfe	0x10-0xfe
Module language mask	ML_MASK	0x00ff

m_attrev
MH_ATTREV

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 3-5. m_attrev Available Attribute and Revision Values

Description	Macro	EditMod
The module is re-entrant (sharable by multiple tasks).	MA_REENT (shifted left to first byte: MA_REENT<<8)	0x80 (shifted left to first byte: 0x8000)
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.	MA_GHOST (shifted left to first byte: MA_GHOST<<8)	0x40 (shifted left to first byte: 0x4000)

Table 3-5. m_attrev Available Attribute and Revision Values (Continued)

Description	Macro	EditMod
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

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 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 3-6. Device Descriptor Data Definition Fields

Field	Description File Macro
<code>dd_port</code>	PORTADDR
<code>dd_lun</code>	LUN
<code>dd_pd_size</code>	PD_SIZE
<code>dd_type</code>	DD_TYPE
<code>dd_mode</code>	DD_MODE
<code>fmgr_name</code>	FMGR_NAME

Table 3-6. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>drv_name</code>	DRV_NAME
<code>dd_class</code>	DD_CLASS

dd_port
PORTADDR**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

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 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

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_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 3-7. dd_type Available Values

Description	Macro	EditMod
Sequential Character File Type	DT_SCF	0x0
Random Block File Type	DT_RBF	0x1
Pipe File Type	DT_PIPE	0x2
Sequential Block File Type	DT_SBF	0x3
Network File Type	DT_NFM	0x4
Compact Disc File Type	DT_CDFM	0x5
User Communication Manager	DT_UCM	0x6
Socket Communication Manager	DT SOCK	0x7

Table 3-7. `dd_type` Available Values (Continued)

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

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_ISIZE | S_IREAD | S_IWRITE`

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](#).

Table 3-8. dd_mode Available Values for File Access Modes

Description	Macro	EditMod
Truncate on open	<code>S_ITRUNC</code>	<code>0x0100</code>
Ensure contiguous file	<code>S_ICONTIG</code>	<code>0x0400</code>
Error if file exists on create	<code>S_IEXCL</code>	<code>0x0400</code>
Create file	<code>S_ICREAT</code>	<code>0x0800</code>

Table 3-8. `dd_mode` Available Values for File Access Modes (Continued)

Description	Macro	EditMod
Append to file	<code>S_IAPPEND</code>	0x1000
Non-sharable	<code>S_ISHARE</code>	0x4000

Table 3-9. `dd_mode` Available Values for File Access Permissions

Description	Macro	EditMod
Mask for permission bits	<code>S_IPRM</code>	0xffff
Owner read	<code>S_IREAD</code>	0x0001
Owner write	<code>S_IWRITE</code>	0x0002
Owner execute	<code>S_IEXEC</code>	0x0004
Search permission	<code>S_ISEARCH</code>	0x0004
Group read	<code>S_IGREAD</code>	0x0010
Group write	<code>S_IGWRITE</code>	0x0020
Group execute	<code>S_IGEXEC</code>	0x0040
Group search	<code>S_IGSEARCH</code>	0x0040
Public read	<code>S_IOREAD</code>	0x0100
Public write	<code>S_IOWRITE</code>	0x0200
Public execute	<code>S_IOEXEC</code>	0x0400
Public search	<code>S_IOSEARCH</code>	0x0400

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

"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).

drv_name
DRV_NAME

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).

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 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 3-10. dd_class Available Values

Description	Macro	EditMod
Sequential access device	DC_SEQ	0x0001
Random access device	DC_RND	0x0002

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 3-11. SCF Description Block Fields

Field	Description File Macro
<code>outdev_name</code>	OUTDEVNAME

outdev_name
OUTDEVNAME

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

Field	Description File Macro
<code>hardware_vector</code>	VECTOR
<code>v_irqlevel</code>	IRQLEVEL
<code>v_priority</code>	PRIORITY
<code>v_pollin</code>	INPUT_TYPE
<code>v_pollout</code>	OUTPUT_TYPE
<code>v_lun</code>	LUN

Table 3-12. Device Descriptor Data Definition Fields (Continued)

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

hardware_vector
VECTOR**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

v_irqlevel
IRQLEVEL

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).

v_priority
PRIORITY**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 3-13. v_pollin Available Values

Description	Macro	EditMod
Interrupt driven	IRQDRIVEN	0x0000
Polled	POLLED	0x0001

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 3-14. v_pollout Available Values

Description	Macro	EditMod
Interrupt driven	IRQDRIVEN	0x0000
Polled	POLLED	0x0001


v_lun
LUN

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

v_irqmask
IRQ_MASK

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

v_maxbuff
MAXBUFF

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

v_insize
INSIZE

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

v_outsize
OUTSIZE**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

v_intr
KYBDINTR

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

SCF/OS-9 Compatible Standard Codes	Macro	EditMod
	C_NULL	0x00
C_REPEAT	CTRL_A	0x01
	CTRL_B	0x02
C_INTR	CTRL_C	0x03
C_REPRINT	CTRL_D	0x04
C_QUIT	CTRL_E	0x05
	CTRL_F	0x06
C_BELL	CTRL_G	0x07

Table 3-15. ASCII Control Character Available Values (Continued)

SCF/OS-9 Compatible Standard Codes	Macro	EditMod
C_BACKSPACE	CTRL_H	0x08
C_TAB	CTRL_I	0x09
C_LINEFEED	CTRL_J	0x0A
	CTRL_K	0x0B
C_FORMFEED	CTRL_L	0x0C
C_CR	CTRL_M	0x0D
	CTRL_N	0x0E
	CTRL_O	0x0F
	CTRL_P	0x10
C_XON	CTRL_Q	0x11
	CTRL_R	0x12
C_XOFF	CTRL_S	0x13
	CTRL_T	0x14
	CTRL_U	0x15
	CTRL_V	0x16
C_PAUSE	CTRL_W	0x17
C_DELLINE	CTRL_X	0x18
	CTRL_Y	0x19
	CTRL_Z	0x1A
	CTRL_SPACE	0x20
	CTRL_COMMA	0x2C
	CTRL_PERIOD	0x2E

Table 3-15. ASCII Control Character Available Values (Continued)

SCF/OS-9 Compatible Standard Codes	Macro	EditMod
	CTRL_SLASH	0x2F
C_EOF		0x1B

v_quit
KYBDQUIT

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](#).

v_psch
KYBDPAUSE

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](#).

v_xon
XON**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

`CTRL_Q`

EditMod

`0x11`**Port Specific Override Value**

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

Available Values

See [Table 3-15](#).


v_xoff
XOFF**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](#).

v_baud
BAUDRATE

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 3-16. `cons_baudrate` Available Values

Description	Macro	EditMod
Hardwire baud rate	HARDWIRE	0x00
50 bits per second (bps)	BAUD50	0x01
75 bps	BAUD75	0x02
110 bps	BAUD110	0x03
134.5 bps	BAUD134P5	0x04
150 bps	BAUD150	0x05
300 bps	BAUD300	0x06
600 pbs	BAUD600	0x07

Table 3-16. `cons_baudrate` Available Values (Continued)

Description	Macro	EditMod
1200 bps	BAUD1200	0x08
1800 bps	BAUD1800	0x09
2000 bps	BAUD2000	0x0A
2400 bps	BAUD2400	0x0B
3600 bps	BAUD3600	0x0C
4800 bps	BAUD4800	0x0D
7200 bps	BAUD7200	0x0E
9600 bps	BAUD9600	0x0F
19,200 bps	BAUD19200	0x10
31,250 bps	BAUD31250	0x11
38,400 bps	BAUD38400	0x12
56,000 bps	BAUD56000	0x13
57,600 bps	BAUD57600	0x14
64,000 bps	BAUD64000	0x15
115,200 bps	BAUD115200	0x16

v_parity
LUPARITY**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 3-17. v_parity Available Values

Description	Macro	EditMod
No parity	NOPARITY	0x00
Odd parity	ODDPARITY	0x01
Even parity	EVENPARITY	0x02
Mark parity	MARKPARITY	0x03
Space parity	SPACEPARITY	0x04

v_stopbits
STOPBITS**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 3-18. `cons_stopbits` Available Values

Description	Macro	EditMod
Stop bit length of 1	ONESTOP	0x0
Stop bit length of 1.5	ONE_5STOP	0x1
Stop bit length of 2	TWO_STOP	0x2

v_wordsize
WORDSIZE

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

WORDSIZE8

EditMod

8

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 3-19. v_wordsize Available Values

Description	Macro	EditMod
5 bits per character	WORDSIZE5	0x5
6 bits per character	WORDSIZE6	0x6
7 bits per character	WORDSIZE7	0x7
8 bits per character	WORDSIZE8	0x8

v_rtsstate
RTSSTATE**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 3-20. v_rtsstate Available Values

Description	Macro	EditMod
rts disabled	RTSDISABLED	0x0
rts enabled	RTSENABLED	0x1

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

Field	Description File Macro
pd_inmap0type	TYPE0x7f
pd_inmap0func_code	FUNC0x7f
pd_inmap0size	SIZE0x7f
pd_inmap0string	STRING0x7f
pd_inmap1type	TYPE0x01
pd_inmap1func_code	FUNC0x01
pd_inmap1size	SIZE0x01
pd_inmap1string	STRING0x01
pd_inmap2type	TYPE0x02
pd_inmap2func_code	FUNC0x02
pd_inmap2size	SIZE0x02

Table 3-21. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>pd_inmap2string</code>	<code>STRING0x02</code>
<code>pd_inmap3type</code>	<code>TYPE0x03</code>
<code>pd_inmap3func_code</code>	<code>FUNC0x03</code>
<code>pd_inmap3size</code>	<code>SIZE0x03</code>
<code>pd_inmap3string</code>	<code>STRING0x03</code>
<code>pd_inmap4type</code>	<code>TYPE0x04</code>
<code>pd_inmap4func_code</code>	<code>FUNC0x04</code>
<code>pd_inmap4size</code>	<code>SIZE0x04</code>
<code>pd_inmap4string</code>	<code>STRING0x04</code>
<code>pd_inmap5type</code>	<code>TYPE0x05</code>
<code>pd_inmap5func_code</code>	<code>FUNC0x05</code>
<code>pd_inmap5size</code>	<code>SIZE0x05</code>
<code>pd_inmap5string</code>	<code>STRING0x05</code>
<code>pd_inmap6type</code>	<code>TYPE0x06</code>
<code>pd_inmap6func_code</code>	<code>FUNC0x06</code>
<code>pd_inmap6size</code>	<code>SIZE0x06</code>
<code>pd_inmap6string</code>	<code>STRING0x06</code>
<code>pd_inmap7type</code>	<code>TYPE0x07</code>
<code>pd_inmap7func_code</code>	<code>FUNC0x07</code>
<code>pd_inmap7size</code>	<code>SIZE0x07</code>
<code>pd_inmap7string</code>	<code>STRING0x07</code>
<code>pd_inmap8type</code>	<code>TYPE0x08</code>

Table 3-21. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>pd_inmap8func_code</code>	<code>FUNC0x08</code>
<code>pd_inmap8size</code>	<code>SIZE0x08</code>
<code>pd_inmap8string</code>	<code>STRING0x08</code>
<code>pd_inmap9type</code>	<code>TYPE0x09</code>
<code>pd_inmap9func_code</code>	<code>FUNC0x09</code>
<code>pd_inmap9size</code>	<code>SIZE0x09</code>
<code>pd_inmap9string</code>	<code>STRING0x09</code>
<code>pd_inmap10type</code>	<code>TYPE0x0a</code>
<code>pd_inmap10func_code</code>	<code>FUNC0x0a</code>
<code>pd_inmap10size</code>	<code>SIZE0x0a</code>
<code>pd_inmap10string</code>	<code>STRING0x0a</code>
<code>pd_inmap11type</code>	<code>TYPE0x0b</code>
<code>pd_inmap11func_code</code>	<code>FUNC0x0b</code>
<code>pd_inmap11size</code>	<code>SIZE0x0b</code>
<code>pd_inmap11string</code>	<code>STRING0x0b</code>
<code>pd_inmap12type</code>	<code>TYPE0x0c</code>
<code>pd_inmap12func_code</code>	<code>FUNC0x0c</code>
<code>pd_inmap12size</code>	<code>SIZE0x0c</code>
<code>pd_inmap12string</code>	<code>STRING0x0c</code>
<code>pd_inmap13type</code>	<code>TYPE0x0d</code>
<code>pd_inmap13func_code</code>	<code>FUNC0x0d</code>
<code>pd_inmap13size</code>	<code>SIZE0x0d</code>

Table 3-21. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>pd_inmap13string</code>	<code>STRING0xd</code>
<code>pd_inmap14type</code>	<code>TYPE0xe</code>
<code>pd_inmap14func_code</code>	<code>FUNC0xe</code>
<code>pd_inmap14size</code>	<code>SIZE0xe</code>
<code>pd_inmap14string</code>	<code>STRING0xe</code>
<code>pd_inmap15type</code>	<code>TYPE0xf</code>
<code>pd_inmap15func_code</code>	<code>FUNC0xf</code>
<code>pd_inmap15size</code>	<code>SIZE0xf</code>
<code>pd_inmap15string</code>	<code>STRING0xf</code>
<code>pd_inmap16type</code>	<code>TYPE0x10</code>
<code>pd_inmap16func_code</code>	<code>FUNC0x10</code>
<code>pd_inmap16size</code>	<code>SIZE0x10</code>
<code>pd_inmap16string</code>	<code>STRING0x10</code>
<code>pd_inmap17type</code>	<code>TYPE0x11</code>
<code>pd_inmap17func_code</code>	<code>FUNC0x11</code>
<code>pd_inmap17size</code>	<code>SIZE0x11</code>
<code>pd_inmap17string</code>	<code>STRING0x11</code>
<code>pd_inmap18type</code>	<code>TYPE0x12</code>
<code>pd_inmap18func_code</code>	<code>FUNC0x12</code>
<code>pd_inmap18size</code>	<code>SIZE0x12</code>
<code>pd_inmap18string</code>	<code>STRING0x12</code>
<code>pd_inmap19type</code>	<code>TYPE0x13</code>

Table 3-21. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>pd_inmap19func_code</code>	<code>FUNC0x13</code>
<code>pd_inmap19size</code>	<code>SIZE0x13</code>
<code>pd_inmap19string</code>	<code>STRING0x13</code>
<code>pd_inmap20type</code>	<code>TYPE0x14</code>
<code>pd_inmap20func_code</code>	<code>FUNC0x14</code>
<code>pd_inmap20size</code>	<code>SIZE0x14</code>
<code>pd_inmap20string</code>	<code>STRING0x14</code>
<code>pd_inmap21type</code>	<code>TYPE0x15</code>
<code>pd_inmap21func_code</code>	<code>FUNC0x15</code>
<code>pd_inmap21size</code>	<code>SIZE0x15</code>
<code>pd_inmap21string</code>	<code>STRING0x15</code>
<code>pd_inmap22type</code>	<code>TYPE0x16</code>
<code>pd_inmap22func_code</code>	<code>FUNC0x16</code>
<code>pd_inmap22size</code>	<code>SIZE0x16</code>
<code>pd_inmap22string</code>	<code>STRING0x16</code>
<code>pd_inmap23type</code>	<code>TYPE0x17</code>
<code>pd_inmap23func_code</code>	<code>FUNC0x17</code>
<code>pd_inmap23size</code>	<code>SIZE0x17</code>
<code>pd_inmap23string</code>	<code>STRING0x17</code>
<code>pd_inmap24type</code>	<code>TYPE0x18</code>
<code>pd_inmap24func_code</code>	<code>FUNC0x18</code>
<code>pd_inmap24size</code>	<code>SIZE0x18</code>

Table 3-21. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>pd_inmap24string</code>	<code>STRING0x18</code>
<code>pd_inmap25type</code>	<code>TYPE0x19</code>
<code>pd_inmap25func_code</code>	<code>FUNC0x19</code>
<code>pd_inmap25size</code>	<code>SIZE0x19</code>
<code>pd_inmap25string</code>	<code>STRING0x19</code>
<code>pd_inmap26type</code>	<code>TYPE0x1a</code>
<code>pd_inmap26func_code</code>	<code>FUNC0x1a</code>
<code>pd_inmap26size</code>	<code>SIZE0x1a</code>
<code>pd_inmap26string</code>	<code>STRING0x1a</code>
<code>pd_inmap27type</code>	<code>TYPE0x1b</code>
<code>pd_inmap27func_code</code>	<code>FUNC0x1b</code>
<code>pd_inmap27size</code>	<code>SIZE0x1b</code>
<code>pd_inmap27string</code>	<code>STRING0x1b</code>
<code>pd_inmap28type</code>	<code>TYPE0x1c</code>
<code>pd_inmap28func_code</code>	<code>FUNC0x1c</code>
<code>pd_inmap28size</code>	<code>SIZE0x1c</code>
<code>pd_inmap28string</code>	<code>STRING0x1c</code>
<code>pd_inmap29type</code>	<code>TYPE0x1d</code>
<code>pd_inmap29func_code</code>	<code>FUNC0x1d</code>
<code>pd_inmap29size</code>	<code>SIZE0x1d</code>
<code>pd_inmap29string</code>	<code>STRING0x1d</code>
<code>pd_inmap30type</code>	<code>TYPE0x1e</code>

Table 3-21. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>pd_inmap30func_code</code>	<code>FUNC0x1e</code>
<code>pd_inmap30size</code>	<code>SIZE0x1e</code>
<code>pd_inmap30string</code>	<code>STRING0x1e</code>
<code>pd_inmap31type</code>	<code>TYPE0x1f</code>
<code>pd_inmap31func_code</code>	<code>FUNC0x1f</code>
<code>pd_inmap31size</code>	<code>SIZE0x1f</code>
<code>pd_inmap31string</code>	<code>STRING0x1f</code>
<code>pd_eorch</code>	<code>EORCH</code>
<code>pd_eofch</code>	<code>EOFCH</code>
<code>pd_tabch</code>	<code>TABCH</code>
<code>pd_bellch</code>	<code>BELLCH</code>
<code>pd_bspch</code>	<code>BSPCH</code>
<code>pd_case</code>	<code>UPC_LOCK</code>
<code>pd_backsp</code>	<code>BSB</code>
<code>pd_delete</code>	<code>LINEDEL</code>
<code>pd_echo</code>	<code>AUTOECHO</code>
<code>pd_alf</code>	<code>AUTOLF</code>
<code>pd_pause</code>	<code>PAGEPAUSE</code>
<code>pd_insm</code>	<code>INSERTMODE</code>
<code>pd_nulls</code>	<code>EOLNULLS</code>
<code>pd_page</code>	<code>PAGESIZE</code>
<code>pd_tabsiz</code>	<code>TABSIZE</code>

pd_inmap0type
TYPE0x7f

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

Control Character is...	Macro	EditMod
removed from the data stream.	IGNORE	0x0
passed on without editing.	PASSTHRU	0x1
removed from the data stream.	EDFUNCTION	0x2

pd_inmap0func_code
FUNC0x7f

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

Description	Macro	EditMod
Move cursor to the left	MOVLEFT	0x00
Move cursor to the right	MOVRIGHT	0x01
Move cursor to the beginning of the line	MOVBEG	0x02
Move cursor to the end of the line	MOVEND	0x03
Reprint the current line to cursor position	REPRINT	0x04
Truncate the line at the cursor position	TRUNCATE	0x05
Delete character to the left	DELCHRL	0x06

Table 3-23. ASCII Control Character Available Values (Continued)

Description	Macro	EditMod
Delete character under the cursor	DELCHRU	0x07
Delete word to the left	DELWRDL	0x08
Delete word to the right	DELWRDR	0x09
Delete the entire line	DELIN	0x0A
Undefined (reserved)	UNDEF1	0x0B
Input mode toggle (type over vs. insert)	MODETOGL	0x0C
Undefined (reserved)	UNDEF2	0x0D
End of record (read only)	ENDOREC	0x0E
End of file	ENDOFIL	0x0F

pd_inmap0size
SIZE0x7f

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

pd_inmap0string
STRING0x7f

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](#).

pd_inmap1func_code
FUNC0x01

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_inmap1 size
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

pd_inmap1string
STRING0x01

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](#).

pd_inmap2func_code
FUNC0x02

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

pd_inmap2string
STRING0x02**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).

pd_inmap3type
TYPE0x03

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](#).

pd_inmap3func_code
FUNC0x03

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](#).

pd_inmap3size
SIZE0x03

EditMod Labels

5-SCF path options

15-'\x03' 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_inmap3string
STRING0x03**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](#).

pd_inmap4func_code

FUNC0x04

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
SIZE0x04

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

pd_inmap4string
STRING0x04**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
TYPE0x05

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-'\x05' 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](#).

pd_inmap6func_code
FUNC0x06

EditMod Labels

5-SCF path options

26-'\x06' 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](#).

pd_inmap6size
SIZE0x06

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

pd_inmap6string
STRING0x06**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](#).

pd_inmap7func_code
FUNC0x07

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

pd_inmap7string
STRING0x07

EditMod Labels

5-SCF path options

32-'\x07' 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_inmap8type
TYPE0x08

EditMod Labels

5-SCF path options

33-'\x08' 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_inmap8func_code
FUNC0x08

EditMod Labels

5-SCF path options

34-'\x08' 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
SIZE0x08

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-'\x08' 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_inmap9type
TYPE0x09

EditMod Labels

5-SCF path options

37-'\x09' 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_inmap9func_code
FUNC0x09

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

pd_inmap9string
STRING0x09

EditMod Labels

5-SCF path options

40-'\x09' 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_inmap10type
TYPE0x0a

EditMod Labels

5-SCF path options

41-'\x0a' 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_inmap10func_code
FUNC0x0a

EditMod Labels

5-SCF path options
42-'\x0a' 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

pd_inmap10string
STRING0x0a

EditMod Labels

5-SCF path options

44-'\x0a' 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-'\x0b' 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_inmap11func_code
FUNC0x0b

EditMod Labels

5-SCF path options

46-'\x0b' 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
SIZE0x0b

EditMod Labels

5-SCF path options

47-'\x0b' 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_inmap11string
STRING0x0b

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
TYPE0x0c

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](#).

pd_inmap12func_code
FUNC0x0c

EditMod Labels

5-SCF path options

50-'\x0c' 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](#).

pd_inmap12size
SIZE0x0c

EditMod Labels

5-SCF path options

51-'\x0c' 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_inmap12string
STRING0x0c

EditMod Labels

5-SCF path options

52-'\x0c' 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_inmap13type
TYPE0x0d

EditMod Labels

5-SCF path options

53-'\x0d' 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_inmap13func_code
FUNC0x0d

EditMod Labels

5-SCF path options

54-'\x0d' editing function code

Description

SCF editing function mapping code for specified character.

Port Generic Default Value

Macro

ENDOREC

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_inmap13size
SIZE0x0d

EditMod Labels

5-SCF path options

55-'\x0d' 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_inmap13string
STRING0x0d

EditMod Labels

5-SCF path options

56-'\x0d' 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-'\x0e' 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_inmap14func_code
FUNC0x0e

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-'\x0e' 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_inmap14string
STRING0x0e

EditMod Labels

5-SCF path options

60-'\x0e' 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_inmap15type
TYPE0x0f

EditMod Labels

5-SCF path options

61-'\x0f' 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_inmap15func_code
FUNC0x0f

EditMod Labels

5-SCF path options

62-'\x0f' 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

pd_inmap15string
STRING0x0f

EditMod Labels

5-SCF path options

64-'\x0f' 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-'\x10' 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_inmap16func_code

FUNC0x10

EditMod Labels

5-SCF path options

66-'\x10' 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-'\x11' 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_inmap17func_code
FUNC0x11

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

pd_inmap17string
STRING0x11

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](#).

pd_inmap18size
SIZE0x12

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

pd_inmap18string
STRING0x12

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).

pd_inmap19type
TYPE0x13

EditMod Labels

5-SCF path options

77-'\x13' 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_inmap19func_code
FUNC0x13

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](#).

pd_inmap19size
SIZE0x13

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](#).

pd_inmap20func_code

FUNC0x14

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](#).

pd_inmap20size
SIZE0x14

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

pd_inmap20string
STRING0x14

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

pd_inmap21string
STRING0x15

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
TYPE0x16

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](#).

pd_inma22func_code
FUNC0x16

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.

pd_inmap22size
SIZE0x16

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

pd_inmap22string
STRING0x16

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](#).

pd_inmap23func_code

FUNC0x17

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

pd_inmap23string
STRING0x17

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).

pd_inmap24type
TYPE0x18

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](#).

pd_inmap24func_code

FUNC0x18

EditMod Labels

5-SCF path options

98-'\x18' 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](#).

pd_inmap24size
SIZE0x18

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

pd_inmap24string
STRING0x18**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
TYPE0x19

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](#).

pd_inmap25func_code

FUNC0x19

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

pd_inmap25string
STRING0x19**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).

pd_inmap26type
TYPE0x1a

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](#).

pd_inmap26func_code
FUNC0x1a

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

MOVBEG

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_inmap26size
SIZE0x1a

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

pd_inmap26string
STRING0x1a

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
TYPE0x1b

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

ENDOFIELD

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](#).

pd_inmap28func_code
FUNC0x1c

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

pd_inmap28string
STRING0x1c

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).

pd_inmap29type
TYPE0x1d

EditMod Labels

5-SCF path options

117-'\x1d' 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_inmap29func_code
FUNC0x1d

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.

pd_inmap29size
SIZE0x1d

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

pd_inmap29string
STRING0x1d

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).

pd_inmap30type
TYPE0x1e

EditMod Labels

5-SCF path options

121-'\x1e' 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_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
SIZE0x1e

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).

pd_inmap31type
TYPE0x1f

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](#).

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

pd_inmap31string
STRING0x1f

EditMod Labels

5-SCF path options

128-'\x1f' 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 3-24. ASCII Control Character Available Values

SCF/OS-9 Compatible Standard Codes	Macro	EditMod
	C_NULL	0x00
C_REPEAT	CTRL_A	0x01
	CTRL_B	0x02
C_INTR	CTRL_C	0x03
C_REPRINT	CTRL_D	0x04
C_QUIT	CTRL_E	0x05
	CTRL_F	0x06
C_BELL	CTRL_G	0x07

Table 3-24. ASCII Control Character Available Values (Continued)

SCF/OS-9 Compatible Standard Codes	Macro	EditMod
C_BACKSPACE	CTRL_H	0x08
C_TAB	CTRL_I	0x09
C_LINEFEED	CTRL_J	0x0A
	CTRL_K	0x0B
C_FORMFEED	CTRL_L	0x0C
C_CR	CTRL_M	0x0D
	CTRL_N	0x0E
	CTRL_O	0x0F
	CTRL_P	0x10
C_XON	CTRL_Q	0x11
	CTRL_R	0x12
C_XOFF	CTRL_S	0x13
	CTRL_T	0x14
	CTRL_U	0x15
	CTRL_V	0x16
C_PAUSE	CTRL_W	0x17
C_DELLINE	CTRL_X	0x18
	CTRL_Y	0x19
	CTRL_Z	0x1A
	CTRL_SPACE	0x20
	CTRL_COMMA	0x2C
	CTRL_PERIOD	0x2E

Table 3-24. ASCII Control Character Available Values (Continued)

SCF/OS-9 Compatible Standard Codes	Macro	EditMod
	CTRL_SLASH	0x2F
C_EOF		0x1B

pd_eofch
EOFCH

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](#).

pd_tabch
TABCH

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](#).

pd_bellch
BELLCH

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](#).

pd_case
UPC_LOCK

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

UPC_LOCK (defined as PLOFF in 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, scf.h, and in Table 3-25.

Table 3-25. pd_case Logic Stage Available Values

Description	Macro	EditMod
Positive logic off - Upper and lower case	PLOFF	0x00
Positive logic on	PLON	0x01
Negative logic off	NLOFF	0x00
Negative logic on	NLON	0x01

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 3-26. pd_backsp Logic Stage Available Values

Description	Macro	EditMod
Positive logic off	PLOFF	0x00
Positive logic on - Destructive backspace	PLON	0x01
Negative logic off	NLOFF	0x00
Negative logic on	NLON	0x01

pd_delete
LINEDEL

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

LINEDEL (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 3-27. `pd_delete` Logic Stage Available Values

Description	Macro	EditMod
Positive logic off - Nondestructive line delete	PLOFF	0x00
Positive logic on - Destructive line delete	PLON	0x01
Negative logic off	NLOFF	0x00
Negative logic on	NLON	0x01

pd_echo
AUTOECHO

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 3-28. `pd_echo` Logic Stage Available Values

Description	Macro	EditMod
Positive logic off - Echo off	<code>PLOFF</code>	<code>0x00</code>
Positive logic on - Echo on	<code>PLON</code>	<code>0x01</code>
Negative logic off	<code>NLOFF</code>	<code>0x00</code>
Negative logic on	<code>NLON</code>	<code>0x01</code>

pd_alf
AUTOLF

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 3-29. `pd_alf` Logic Stage Available Values

Description	Macro	EditMod
Positive logic off - Auto linefeed off	PLOFF	0x00
Positive logic on - Auto linefeed on	PLON	0x01
Negative logic off	NLOFF	0x00
Negative logic on	NLON	0x01

pd_pause
PAGEPAUSE

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 3-30. `pd_pause` Logic Stage Available Values

Description	Macro	EditMod
Positive logic off - Auto linefeed off	PLOFF	0x00
Positive logic on - Auto linefeed on	PLON	0x01
Negative logic off	NLOFF	0x00
Negative logic on	NLON	0x01

pd_insm
INSERTMODE

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

INSERTMODE (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 3-31. `pd_insmLogic` Stage Available Values

Description	Macro	EditMod
Positive logic off - Insert mode off	PLOFF	0x00
Positive logic on - Insert mode on	PLON	0x01
Negative logic off	NLOFF	0x00
Negative logic on	NLON	0x01

pd_nulls
EOLNULLS

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

pd_page
PAGESIZE

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

pd_tabsiz
TABSIZ

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

4

SBF Device Descriptors



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.

Figure 4-1. Directory Location for Modifying SBF Device Descriptors

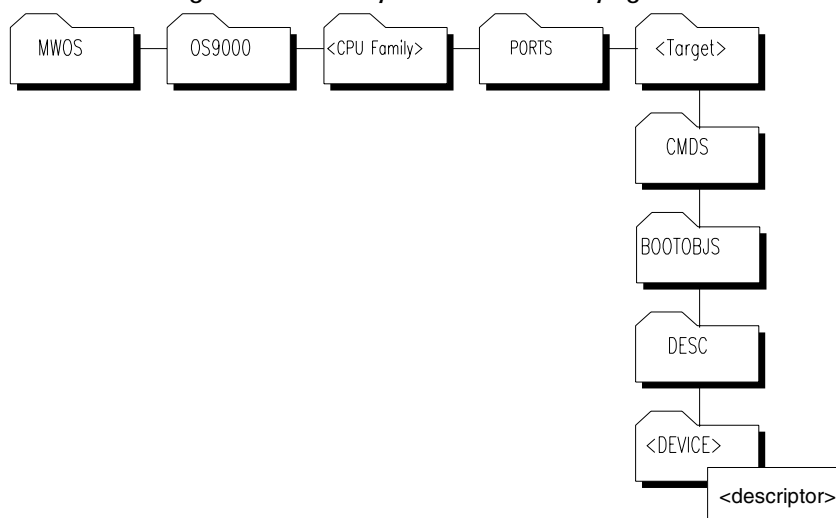
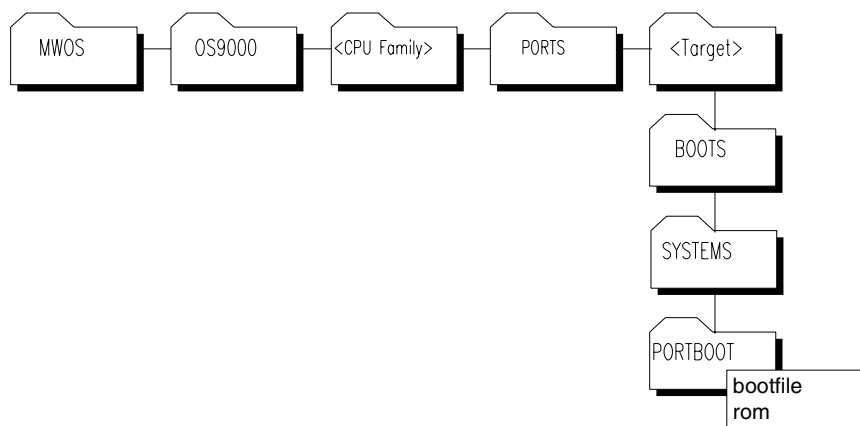


Figure 4-2. Directory Location for Modifying Low-Level Boot Images



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/BOOTOBS/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 |

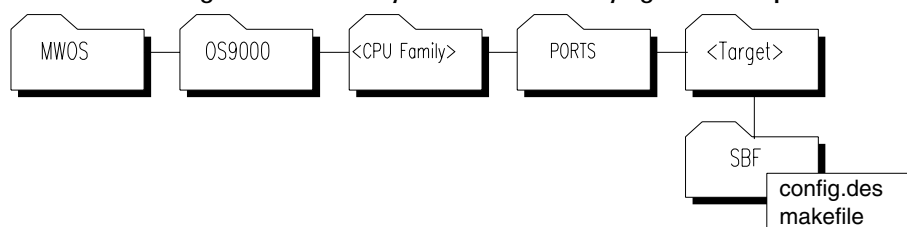
```
Which? [?/1-19/p/t/a/w/q] w
```

```
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



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:

```
#define <macro> <value>
```

6. Save the changes and rebuild the SBF device descriptors, entering the following command in the SBF/<DEVICE>/DESC directory:

```
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 4-1. Module Header Fields

Field	Description File Macro
<code>_m_group</code>	MH_GROUP
<code>_m_user</code>	MH_USER
<code>mod_name</code>	MH_NAME
<code>m_access</code>	MH_ACCESS
<code>m_tylan</code>	MH_TYLAN
<code>m_attrev</code>	MH_ATTREV
<code>m_edit</code>	MH_EDITION

_m_group
MH_GROUP

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

_m_user
MH_USER

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

mod_name
MH_NAME

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`).

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 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 4-2. m_access Available Values

Description	Macro	EditMod
Read permission by owner	MP_OWNER_READ	0x0001
Write permission by owner	MP_OWNER_WRITE	0x0002
Execute permission by owner	MP_OWNER_EXEC	0x0004
Owner permission mask	MP_OWNER_MASK	0x000f
Read permission by group	MP_GROUP_READ	0x0010
Write permission by group	MP_GROUP_WRITE	0x0020
Execute permission by group	MP_GROUP_EXEC	0x0040

Table 4-2. `m_access` Available Values (Continued)

Description	Macro	EditMod
Group permission mask	MP_GROUP_MASK	0x00f0
Read permission by world	MP_WORLD_READ	0x0100
Write permission by world	MP_WORLD_WRITE	0x0200
Execute permission by world	MP_WORLD_EXEC	0x0400
World permission mask	MP_WORLD_MASK	0x0f00
All permissions for owner, group, and world	MP_WORLD_ACCESS	0x0777
System permission mask	MP_SYSTM_MASK	0xf000

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 4-3. m_tylan Available Module Type Values

Description	Macro	EditMod
Not used (wildcard value in system calls)	MT_ANY	0x0000
Program module	MT_PROGRAM	0x0001
Subroutine module	MT_SUBROUT	0x0002
Multi-module (reserved for future use)	MT_MULTI	0x0003
Data module	MT_DATA	0x0004
Configuration data block data module	MT_CDBDATA	0x0005
Reserved for future use	0xb-0xa	0xb-0xa

Table 4-3. `m_tylan` Available Module Type Values (Continued)

Description	Macro	EditMod
User trap library	MT_TRAPLIB	0x000b
System module	MT_SYSTEM	0x000c
File manager module	MT_FILEMAN	0x000d
Physical device driver	MT_DEVDVR	0x000e
Device descriptor module	MT_DEVDESC	0x000f
User definable	0x10-0xfe	0x10-0xfe
Module type mask	MT_MASK	0xff00

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

Description	Macro	EditMod
Unspecified language (wildcard in system calls)	ML_ANY	0x0
Machine language	ML_OBJECT	0x1
Basic I-code (reserved for future use)	ML_ICODE	0x2
Pascal P-code (reserved for future use)	ML_PCODE	0x3
C I-code (reserved for future use)	ML_CCODE	0x4
Cobol I-code (reserved for future use)	ML_CBLCODE	0x5
Fortran	ML_FRTNCODE	0x6
Reserved for future use	0x7-0xf	0x7-0xf
User-definable	0x10-0xfe	0x10-0xfe
Module language mask	ML_MASK	0x00ff

m_attrev
MH_ATTREV

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 4-5. m_attrev Available Attribute and Revision Values

Description	Macro	EditMod
The module is re-entrant (sharable by multiple tasks).	MA_REENT (shifted left to first byte: MA_REENT<<8)	0x80 (shifted left to first byte: 0x8000)
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.	MA_GHOST (shifted left to first byte: MA_GHOST<<8)	0x40 (shifted left to first byte: 0x4000)

Table 4-5. `m_attrrev` Available Attribute and Revision Values (Continued)

Description	Macro	EditMod
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 4-6. Device Descriptor Data Definition Fields

Field	Description File Macro
dd_port	PORTADDR
dd_lun	LUN
dd_pd_size	PD_SIZE
dd_type	DD_TYPE
dd_mode	DD_MODE
dd_port	MFGR_NAME

Table 4-6. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>drv_name</code>	DRV_NAME
<code>dd_class</code>	DD_CLASS

dd_port
PORTADDR

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 SBF/<DEVICE>/DESC/config.des ([Figure 4-3](#)).

Available Values

0 to 4294967295


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

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_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 4-7. dd_type Available Values

Description	Macro	EditMod
Sequential Character File Type	DT_SCF	0x0
Random Block File Type	DT_RBF	0x1
Pipe File Type	DT_PIPE	0x2
Sequential Block File Type	DT_SBF	0x3
Network File Type	DT_NFM	0x4
Compact Disc File Type	DT_CDFM	0x5
User Communication Manager	DT_UCM	0x6
Socket Communication Manager	DT SOCK	0x7

Table 4-7. `dd_type` Available Values (Continued)

Description	Macro	EditMod
Pseudo-Keyboard Manager	DT_PTTY	0x8
Graphics File Manager	DT_GFM	0x9
PC-DOS File Manager	DT_PCF	0xa
Non-volatile RAM File Manager	DT_NRF	0xb
ISDN File Manager	DT_ISDN	0xc
MPFM File Manager	DT_MPFM	0xd
Real-Time Network File Manager	DT_RTNFM	0xe
Serial Protocol File Manager	DT_SPF	0xf
Inet File Manager	DT_INET	0xa0
Reserved for Microware Use Only	17-127	0xa1-0x7f

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 `SBF/<DEVICE>/DESC/config.des` ([Figure 4-3](#)).

Available Values

The file access modes are defined in the header file, `modes.h`, and located in [Table 4-8](#). The file access permission values are defined in the header file `modes.h` and in [Table 4-9](#).

Table 4-8. `dd_mode` Available Values for File Access Modes

Description	Macro	EditMod
Truncate on open	<code>S_ITRUNC</code>	<code>0x0100</code>
Ensure contiguous file	<code>S_ICONTIG</code>	<code>0x0400</code>
Error if file exists on create	<code>S_IEXCL</code>	<code>0x0400</code>
Create file	<code>S_ICREAT</code>	<code>0x0800</code>

Table 4-8. `dd_mode` Available Values for File Access Modes (Continued)

Description	Macro	EditMod
Append to file	<code>S_IAPPEND</code>	0x1000
Non-sharable	<code>S_ISHARE</code>	0x4000

Table 4-9. `dd_mode` Available Values for File Access Permissions

Description	Macro	EditMod
Mask for permission bits	<code>S_IPRM</code>	0xffff
Owner read	<code>S_IREAD</code>	0x0001
Owner write	<code>S_IWRITE</code>	0x0002
Owner execute	<code>S_IEXEC</code>	0x0004
Search permission	<code>S_ISEARCH</code>	0x0004
Group read	<code>S_IGREAD</code>	0x0010
Group write	<code>S_IGWRITE</code>	0x0020
Group execute	<code>S_IGEXEC</code>	0x0040
Group search	<code>S_IGSEARCH</code>	0x0040
Public read	<code>S_IOREAD</code>	0x0100
Public write	<code>S_IOWRITE</code>	0x0200
Public execute	<code>S_IOEXEC</code>	0x0400
Public search	<code>S_IOSEARCH</code>	0x0400

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).

drv_name
DRV_NAME

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
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 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 4-10. dd_class Available Values

Description	Macro	EditMod
Sequential access device	DC_SEQ	0x0001
Random access device	DC_RND	0x0002

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 4-11. SBF Path Options Fields

Field	Description File Macro
<code>pd_blksize</code>	BLKSIZE
<code>pd_flags</code>	FLAGS
<code>pd_dmamode</code>	DMAMODE
<code>pd_sci_id</code>	SCSIID
<code>pd_scsilun</code>	SCSILUN

pd_blksize
BLKSIZE

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

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 `sbfd.h`, and in Table 4-12.

Table 4-12. SBF Compatibility Flag

Description	Macro	EditMod
Rewind tape on close	DEV_REWIND_FLG	0x0001
Erase to end after writing	DEV_ERASE_FLG	0x0002
Take drive off-line on close	DEV_OFFLINE_FLG	0x0004
Device can skip backwards	DEV_SKIPBACK_FLG	0x0008

pd_dmamode
DMAMODE

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

pd_sci_id
SCSIID

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 4-13. SBF Logical Unit Static Storage Fields

Field	Description File Macro
<code>sbf_vector</code>	VECTOR
<code>sbf_irqlevel</code>	IRQLEVEL
<code>sbf_priority</code>	PRIORITY
<code>sbf_dflag</code>	DRIVE_FLAG

sbf_vector
VECTOR**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
IRQLEVEL

EditMod Labels

4-SBF logical unit status

2-irq level

Description

This is the hardware priority 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_priority
PRIORITY**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

sbfdflag
DRIVE_FLAG**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 ValuesDrive flag values are defined in the header file `sbfd.h`, and in Table 4-14.**Table 4-14. SBF Drive Flag**

Description	Macro	EditMod
Read is in progress on device	DFLG_READFLAG	0x0001
Write is in progress on device	DFLG_WRITEFLAG	0x0002
Driver is using the device	DFLG_DRIVEBUSY	0x0004
Drive is at EOF	DFLG_EOFFLAG	0x0008



5

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.

Figure 5-1. Directory Location for Modifying RBF Device Descriptors

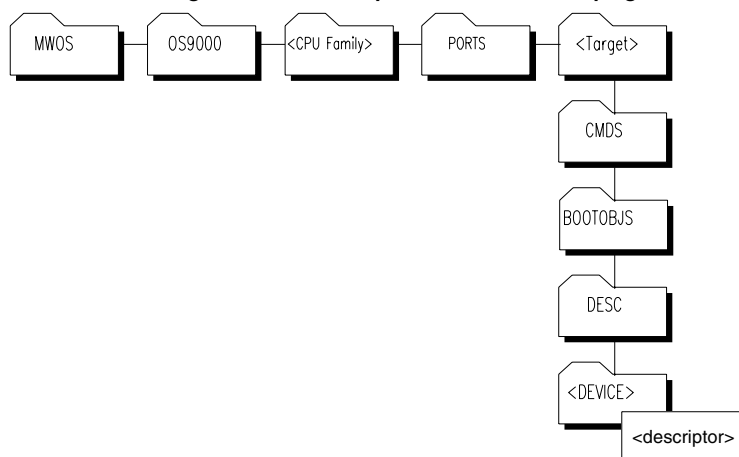
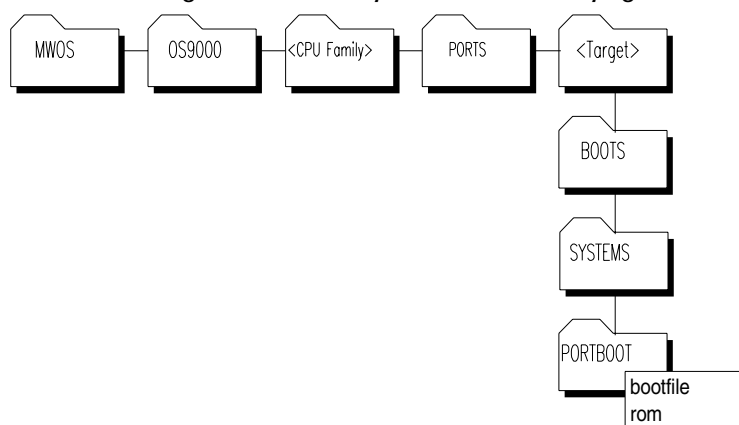


Figure 5-2. Directory Location for Modifying Low-Level Boot Images



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 CMD5/BOOTOBJ5/DESC/<DEVICE> directory (see [Figure 5-1](#)).
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 5-2](#)).
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 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 | |

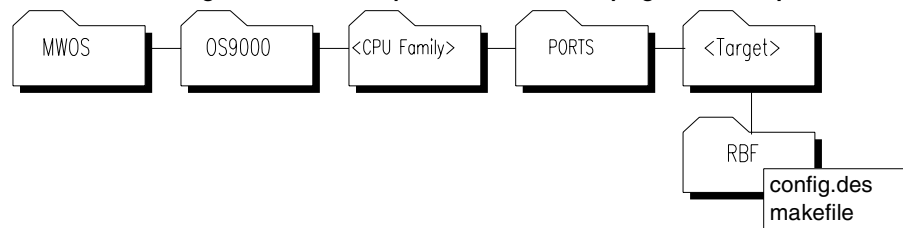
```
Which? [?/1-19/p/t/a/w/q] w
```

```
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 5-1. Module Header Fields

Field	Description File Macro
<code>_m_group</code>	MH_GROUP
<code>_m_user</code>	MH_USER
<code>mod_name</code>	MH_NAME
<code>m_access</code>	MH_ACCESS
<code>m_tylan</code>	MH_TYLAN
<code>m_attrev</code>	MH_ATTREV
<code>m_edit</code>	MH_EDITION

_m_group
MH_GROUP

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

_m_user
MH_USER

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

mod_name
MH_NAME

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).

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 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 5-2. m_access Available Values

Description	Macro	EditMod
Read permission by owner	MP_OWNER_READ	0x0001
Write permission by owner	MP_OWNER_WRITE	0x0002
Execute permission by owner	MP_OWNER_EXEC	0x0004
Owner permission mask	MP_OWNER_MASK	0x000f
Read permission by group	MP_GROUP_READ	0x0010
Write permission by group	MP_GROUP_WRITE	0x0020
Execute permission by group	MP_GROUP_EXEC	0x0040

Table 5-2. `m_access` Available Values (Continued)

Description	Macro	EditMod
Group permission mask	MP_GROUP_MASK	0x00f0
Read permission by world	MP_WORLD_READ	0x0100
Write permission by world	MP_WORLD_WRITE	0x0200
Execute permission by world	MP_WORLD_EXEC	0x0400
World permission mask	MP_WORLD_MASK	0x0f00
All permissions for owner, group, and world	MP_WORLD_ACCESS	0x0777
System permission mask	MP_SYSTM_MASK	0xf000

m_tylan
MH_TYLAN

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 RBF/<DEVICE>/DESC/config.des ([Figure 5-3](#)).

Available Values

Module type values and language codes are located in the header file, `module.h`, and are listed in [Table 5-3](#) and [Table 5-4](#).

Table 5-3. **m_tylan** Available Module Type Values

Description	Macro	EditMod
Not used (wildcard value in system calls)	MT_ANY	0x0000
Program module	MT_PROGRAM	0x0001
Subroutine module	MT_SUBROUT	0x0002
Multi-module (reserved for future use)	MT_MULTI	0x0003
Data module	MT_DATA	0x0004
Configuration data block data module	MT_CDBDATA	0x0005
Reserved for future use	0xb-0xa	0xb-0xa

Table 5-3. `m_tylan` Available Module Type Values (Continued)

Description	Macro	EditMod
User trap library	MT_TRAPLIB	0x000b
System module	MT_SYSTEM	0x000c
File manager module	MT_FILEMAN	0x000d
Physical device driver	MT_DEVDVR	0x000e
Device descriptor module	MT_DEVDESC	0x000f
User definable	0x10-0xfe	0x10-0xfe
Module type mask	MT_MASK	0xff00

Table 5-4. `m_tylan` Available Language Code Values

Description	Macro	EditMod
Unspecified language (wildcard in system calls)	ML_ANY	0x0
Machine language	ML_OBJECT	0x1
Basic I-code (reserved for future use)	ML_ICODE	0x2
Pascal P-code (reserved for future use)	ML_PCODE	0x3
C I-code (reserved for future use)	ML_CCODE	0x4
Cobol I-code (reserved for future use)	ML_CBLCODE	0x5
Fortran	ML_FRTNCODE	0x6
Reserved for future use	0x7-0xf	0x7-0xf
User-definable	0x10-0xfe	0x10-0xfe
Module language mask	ML_MASK	0x00ff

m_attrev
MH_ATTREV

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

Description	Macro	EditMod
The module is re-entrant (sharable by multiple tasks).	MA_REENT (shifted left to first byte: MA_REENT<<8)	0x80 (shifted left to first byte: 0x8000)
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.	MA_GHOST (shifted left to first byte: MA_GHOST<<8)	0x40 (shifted left to first byte: 0x4000)

Table 5-5. `m_attrrev` Available Attribute and Revision Values (Continued)

Description	Macro	EditMod
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 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 5-6. Device Descriptor Data Definition Fields

Field	Description File Macro
dd_port	PORTADDR
dd_lun	LUN
dd_pd_size	PD_SIZE
dd_type	DD_TYPE
dd_mode	DD_MODE
dd_port	MFGR_NAME

Table 5-6. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>drv_name</code>	DRV_NAME
<code>dd_class</code>	DD_CLASS

dd_port
PORTADDR

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

```
#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 5-7. dd_type Available Values

Description	Macro	EditMod
Sequential Character File Type	DT_SCF	0x0
Random Block File Type	DT_RBF	0x1
Pipe File Type	DT_PIPE	0x2
Sequential Block File Type	DT_SBF	0x3
Network File Type	DT_NFM	0x4
Compact Disc File Type	DT_CDFM	0x5
User Communication Manager	DT_UCM	0x6
Socket Communication Manager	DT SOCK	0x7

Table 5-7. `dd_type` Available Values (Continued)

Description	Macro	EditMod
Pseudo-Keyboard Manager	DT_PTTY	0x8
Graphics File Manager	DT_GFM	0x9
PC-DOS File Manager	DT_PCF	0xa
Non-volatile RAM File Manager	DT_NRF	0xb
ISDN File Manager	DT_ISDN	0xc
MPFM File Manager	DT_MPFM	0xd
Real-Time Network File Manager	DT_RTNFM	0xe
Serial Protocol File Manager	DT_SPF	0xf
Inet File Manager	DT_INET	0xa0
Reserved for Microware Use Only	17-127	0xa1-0x7f

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 5-8. `dd_mode` Available Values for File Access Modes

Description	Macro	EditMod
Truncate on open	<code>S_ITRUNC</code>	<code>0x0100</code>
Ensure contiguous file	<code>S_ICONTIG</code>	<code>0x0400</code>
Error if file exists on create	<code>S_IEXCL</code>	<code>0x0400</code>
Create file	<code>S_ICREAT</code>	<code>0x0800</code>

Table 5-8. `dd_mode` Available Values for File Access Modes (Continued)

Description	Macro	EditMod
Append to file	<code>S_IAPPEND</code>	0x1000
Non-sharable	<code>S_ISHARE</code>	0x4000

Table 5-9. `dd_mode` Available Values for File Access Permissions

Description	Macro	EditMod
Mask for permission bits	<code>S_IPRM</code>	0xffff
Owner read	<code>S_IREAD</code>	0x0001
Owner write	<code>S_IWRITE</code>	0x0002
Owner execute	<code>S_IEXEC</code>	0x0004
Search permission	<code>S_ISEARCH</code>	0x0004
Group read	<code>S_IGREAD</code>	0x0010
Group write	<code>S_IGWRITE</code>	0x0020
Group execute	<code>S_IGEXEC</code>	0x0040
Group search	<code>S_IGSEARCH</code>	0x0040
Public read	<code>S_IOREAD</code>	0x0100
Public write	<code>S_IOWRITE</code>	0x0200
Public execute	<code>S_IOEXEC</code>	0x0400
Public search	<code>S_IOSEARCH</code>	0x0400

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

"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).

drv_name
DRV_NAME

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
DD_CLASS

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 5-10. dd_class Available Values

Description	Macro	EditMod
Sequential access device	DC_SEQ	0x0001
Random access device	DC_RND	0x0002

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 5-11. RBF Path Option Fields

Field	Description File Macro
<code>pd_sid</code>	SIDES
<code>pd_vfy</code>	VERIFY
<code>pd_format</code>	FORMAT
<code>pd_cyl</code>	CYLNDRS
<code>pd_blk</code>	BLKSTRK
<code>pd_t0b</code>	BLKSTRK0
<code>pd_sas</code>	SEGSIZE
<code>pd_ilv</code>	INTRLV
<code>pd_toffs</code>	TRKOFFS
<code>pd_boffs</code>	BLKOFFS
<code>pd_trys</code>	TRYS
<code>pd_bsize</code>	BLKSIZE
<code>pd_cntl</code>	CONTROL
<code>pd_wpc</code>	PRECOMP
<code>pd_rwr</code>	REDWRITE
<code>pd_park</code>	PARK
<code>pd_lsnoffs</code>	LSNOFFS
<code>pd_xfersize</code>	XFERSIZE

pd_sid
SIDES

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

pd_vfy
VERIFY**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 5-12. `pd_vfy` Available Values

Description	Macro	EditMod
Verify disk write	0	0x0
No verification	1	0x01

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

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

Description	Macro	EditMod
Track 0 is double density.	FMT_DBLTRK0	0x0001
Device is double bit density.	FMT_DBLBITDNS	0x0002
Device is double track density.	FMT_DBLTRKDNS	0x0004
Device is double sided.	FMT_DBLSIDE	0x0008
Drive is eight inch.	FMT_EIGHTINCH	0x0010
Drive is five inch.	FMT_FIVEINCH	0x0020
Drive is three inch.	FMT_THREEINCH	0x0040
Device is high density.	FMT_HIGHDENS	0x1000

Table 5-13. `pd_format` Available Values (Continued)

Description	Macro	EditMod
Device is standard format.	FMT_STDFMT	0x2000
Media can be removed.	FMT_REMOVABLE	0x4000
Device is a hard disk.	FMT_HARDISK	0x8000

pd_cyl
CYLNDRS

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

pd_blk
BLKSTRK

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

pd_t0b
BLKSTRK0

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

pd_sas
SEGSIZE

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


pd_ilv
INTRLV

EditMod Labels

3-RBF 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 RBF/<DEVICE>/DESC/config.des ([Figure 5-3](#)).

Available Values

-2147483648 to 2147483647

pd_toffs
TRKOFFS

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

pd_boffs
BLKOFFS

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

pd_trys
TRYs

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

pd_bsize
BLKSIZE

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 5-14. pd_ctrl Available Values

Description	Macro	EditMod
Disable formatting of the device	CTRL_FMTDIS	0x0
Device is capable of multi-sector transfers	CTRL_MULTI	0x1
Device size can be obtained from device	CTRL_AUTOSIZE	0x2
Device requires only one format command	CTRL_FMTENTIRE	0x3
Device needs a full track buffer for format	CTRL_TRKWRITE	0x4

pd_wpc
PRECOMP

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

pd_park
PARK

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

pd_lsnoffs
LSNOFFS

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

pd_xfersize
XFERSIZE

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 5-15. RBF Logical Unit Static Storage Fields

Field	Description File Macro
<code>v_vector</code>	VECTOR
<code>v_irqlevel</code>	IRQLEVEL
<code>v_priority</code>	PRIORITY

v_vector
VECTOR**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

v_irqlevel
IRQLEVEL

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

v_priority
PRIORITY

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 5-16. RBF Logical Unit Options Fields

Field	Description File Macro
lu_stp	STEP
lu_tfm	DMAMODE
lu_lun	SCSILUN
lu_ctrlrid	CTRLRID
lu_totcyls	TOTCYLS

lu_stp
STEP

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 5-17. `lu_stp` Available Values

Description	Macro	EditMod
30 millisecond step rate	STEP_30MS	0x00
20 millisecond step rate	STEP_20MS	0x01
12 millisecond step rate	STEP_12MS	0x02
6 millisecond step rate	STEP_6MS	0x03

lu_tfm
DMAMODE

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

lu_ctrlrid
CTRLRID

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

lu_totcyls
TOTCYLS

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

6

PCF Device Descriptors



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.

Figure 6-1. Directory Location for Modifying PCF Device Descriptors

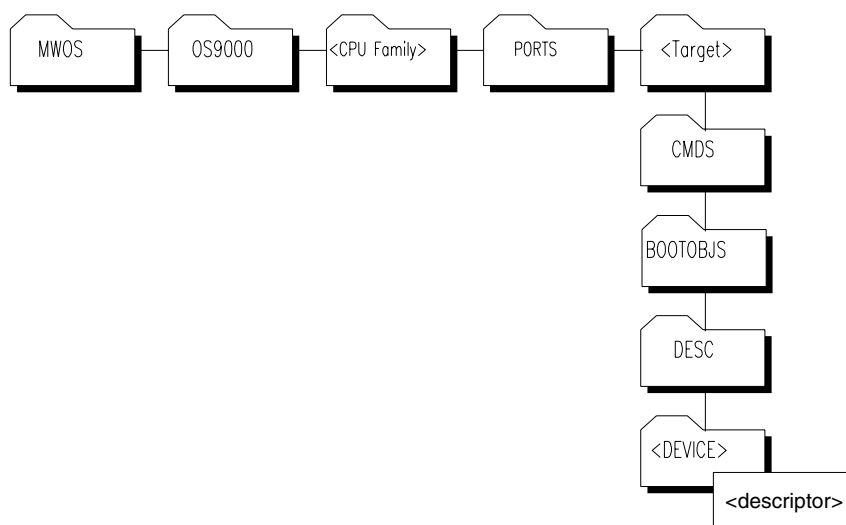
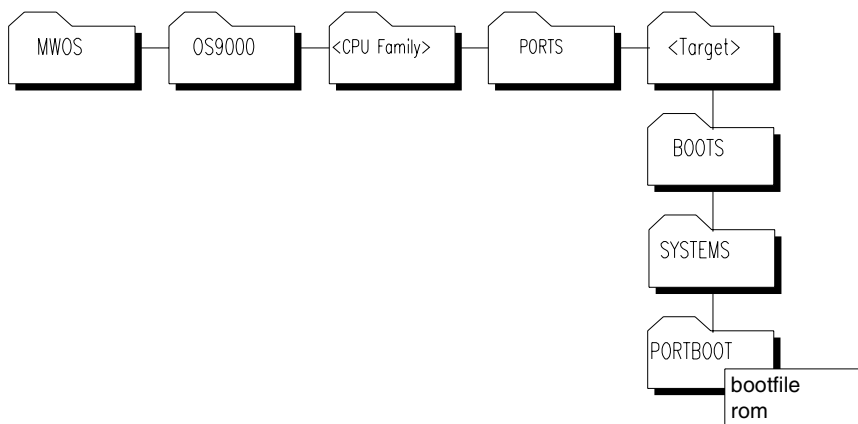


Figure 6-2. Directory Location for Modifying Low-Level Boot Images



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/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 | |

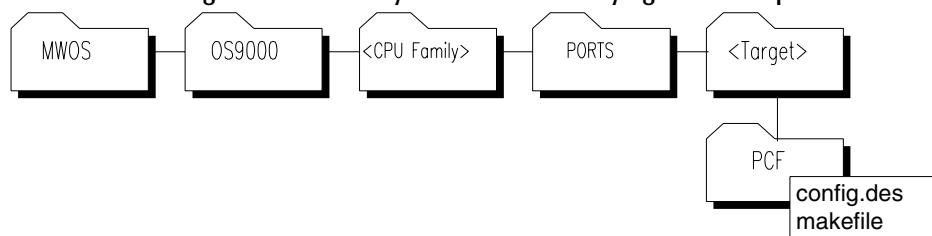
```
Which? [?/1-19/p/t/a/w/q] w
```

```
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 6-1. Module Header Fields

Field	Description File Macro
<code>_m_group</code>	MH_GROUP
<code>_m_user</code>	MH_USER
<code>mod_name</code>	MH_NAME
<code>m_access</code>	MH_ACCESS
<code>m_tylan</code>	MH_TYLAN
<code>m_attrev</code>	MH_ATTREV
<code>m_edit</code>	MH_EDITION

_m_group
MH_GROUP

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 PCF/<DEVICE>/DESC/config.des ([Figure 6-3](#)).

Available Values

0 to 65535

_m_user
MH_USER

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

mod_name
MH_NAME

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).

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 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 6-2. m_access Available Values

Description	Macro	EditMod
Read permission by owner	MP_OWNER_READ	0x0001
Write permission by owner	MP_OWNER_WRITE	0x0002
Execute permission by owner	MP_OWNER_EXEC	0x0004
Owner permission mask	MP_OWNER_MASK	0x000f
Read permission by group	MP_GROUP_READ	0x0010
Write permission by group	MP_GROUP_WRITE	0x0020
Execute permission by group	MP_GROUP_EXEC	0x0040

Table 6-2. `m_access` Available Values (Continued)

Description	Macro	EditMod
Group permission mask	MP_GROUP_MASK	0x00f0
Read permission by world	MP_WORLD_READ	0x0100
Write permission by world	MP_WORLD_WRITE	0x0200
Execute permission by world	MP_WORLD_EXEC	0x0400
World permission mask	MP_WORLD_MASK	0x0f00
All permissions for owner, group, and world	MP_WORLD_ACCESS	0x0777
System permission mask	MP_SYSTM_MASK	0xf000

m_tylan
MH_TYLAN

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 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

Description	Macro	EditMod
Not used (wildcard value in system calls)	MT_ANY	0x0000
Program module	MT_PROGRAM	0x0001
Subroutine module	MT_SUBROUT	0x0002
Multi-module (reserved for future use)	MT_MULTI	0x0003
Data module	MT_DATA	0x0004
Configuration data block data module	MT_CDBDATA	0x0005
Reserved for future use	0xb-0xa	0xb-0xa

Table 6-3. `m_tylan` Available Module Type Values (Continued)

Description	Macro	EditMod
User trap library	MT_TRAPLIB	0x000b
System module	MT_SYSTEM	0x000c
File manager module	MT_FILEMAN	0x000d
Physical device driver	MT_DEVDVR	0x000e
Device descriptor module	MT_DEVDESC	0x000f
User definable	0x10-0xfe	0x10-0xfe
Module type mask	MT_MASK	0xff00

Table 6-4. `m_tylan` Available Language Code Values

Description	Macro	EditMod
Unspecified language (wildcard in system calls)	ML_ANY	0x0
Machine language	ML_OBJECT	0x1
Basic I-code (reserved for future use)	ML_ICODE	0x2
Pascal P-code (reserved for future use)	ML_PCODE	0x3
C I-code (reserved for future use)	ML_CCODE	0x4
Cobol I-code (reserved for future use)	ML_CBLCODE	0x5
Fortran	ML_FRTNCODE	0x6
Reserved for future use	0x7-0xf	0x7-0xf
User-definable	0x10-0xfe	0x10-0xfe
Module language mask	ML_MASK	0x00ff

m_attrev
MH_ATTREV

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 6-5. m_attrev Available Attribute and Revision Values

Description	Macro	EditMod
The module is re-entrant (sharable by multiple tasks).	MA_REENT (shifted left to first byte: MA_REENT<<8)	0x80 (shifted left to first byte: 0x8000)
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.	MA_GHOST (shifted left to first byte: MA_GHOST<<8)	0x40 (shifted left to first byte: 0x4000)

Table 6-5. `m_attrrev` Available Attribute and Revision Values (Continued)

Description	Macro	EditMod
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 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 6-6. Device Descriptor Data Definition Fields

Field	Description File Macro
<code>dd_port</code>	PORTADDR
<code>dd_lun</code>	LUN
<code>dd_pd_size</code>	PD_SIZE
<code>dd_type</code>	DD_TYPE
<code>dd_mode</code>	DD_MODE
<code>dd_port</code>	MFGR_NAME

Table 6-6. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>drv_name</code>	DRV_NAME
<code>dd_class</code>	DD_CLASS

dd_port
PORTADDR

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 PCF/<DEVICE>/DESC/config.des ([Figure 6-3](#)).

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

```
#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
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 PCF/<DEVICE>/DESC/config.des ([Figure 6-3](#)).

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_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 6-7. dd_type Available Values

Description	Macro	EditMod
Sequential Character File Type	DT_SCF	0x0
Random Block File Type	DT_RBF	0x1
Pipe File Type	DT_PIPE	0x2
Sequential Block File Type	DT_SBF	0x3
Network File Type	DT_NFM	0x4
Compact Disc File Type	DT_CDFM	0x5
User Communication Manager	DT_UCM	0x6
Socket Communication Manager	DT SOCK	0x7

Table 6-7. `dd_type` Available Values (Continued)

Description	Macro	EditMod
Pseudo-Keyboard Manager	DT_PTTY	0x8
Graphics File Manager	DT_GFM	0x9
PC-DOS File Manager	DT_PCF	0xa
Non-volatile RAM File Manager	DT_NRF	0xb
ISDN File Manager	DT_ISDN	0xc
MPFM File Manager	DT_MPFM	0xd
Real-Time Network File Manager	DT_RTNFM	0xe
Serial Protocol File Manager	DT_SPF	0xf
Inet File Manager	DT_INET	0xa0
Reserved for Microware Use Only	17-127	0xa1-0x7f

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 `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

Description	Macro	EditMod
Truncate on open	<code>S_ITRUNC</code>	<code>0x0100</code>
Ensure contiguous file	<code>S_ICONTIG</code>	<code>0x0400</code>
Error if file exists on create	<code>S_IEXCL</code>	<code>0x0400</code>
Create file	<code>S_ICREAT</code>	<code>0x0800</code>

Table 6-8. `dd_mode` Available Values for File Access Modes (Continued)

Description	Macro	EditMod
Append to file	<code>S_IAPPEND</code>	0x1000
Non-sharable	<code>S_ISHARE</code>	0x4000

Table 6-9. `dd_mode` Available Values for File Access Permissions

Description	Macro	EditMod
Mask for permission bits	<code>S_IPRM</code>	0xffff
Owner read	<code>S_IREAD</code>	0x0001
Owner write	<code>S_IWRITE</code>	0x0002
Owner execute	<code>S_IEXEC</code>	0x0004
Search permission	<code>S_ISEARCH</code>	0x0004
Group read	<code>S_IGREAD</code>	0x0010
Group write	<code>S_IGWRITE</code>	0x0020
Group execute	<code>S_IGEXEC</code>	0x0040
Group search	<code>S_IGSEARCH</code>	0x0040
Public read	<code>S_IOREAD</code>	0x0100
Public write	<code>S_IOWRITE</code>	0x0200
Public execute	<code>S_IOEXEC</code>	0x0400
Public search	<code>S_IOSEARCH</code>	0x0400

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

"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).

drv_name
DRV_NAME

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
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 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 6-10. dd_class Available Values

Description	Macro	EditMod
Sequential access device	DC_SEQ	0x0001
Random access device	DC_RND	0x0002

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

Field	Description File Macro
<code>pd_sid</code>	SIDES
<code>pd_vfy</code>	VERIFY
<code>pd_format</code>	FORMAT
<code>pd_cyl</code>	CYLNDRS
<code>pd_blk</code>	BLKSTRK
<code>pd_t0b</code>	BLKSTRK0
<code>pd_sas</code>	SEGSIZE
<code>pd_ilv</code>	INTRLV
<code>pd_toffs</code>	TRKOFFS
<code>pd_boffs</code>	BLKOFFS
<code>pd_trys</code>	TRYS
<code>pd_bsize</code>	BLKSIZE
<code>pd_cntl</code>	CONTROL
<code>pd_wpc</code>	PRECOMP
<code>pd_rwr</code>	REDWRITE
<code>pd_park</code>	PARK
<code>pd_lsnoffs</code>	LSNOFFS
<code>pd_xfersize</code>	XFERSIZE

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

pd_vfy
VERIFY**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

Description	Macro	EditMod
Verify disk write	0	0x0
No verification	1	0x01

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

FMT_STDFMT + FMT_DBLBITDNS + 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 6-13. pd_format Available Values

Description	Macro	EditMod
Track 0 is double density.	FMT_DBLTRK0	0x0001
Device is double bit density.	FMT_DBLBITDNS	0x0002
Device is double track density.	FMT_DBLTRKDNS	0x0004
Device is double sided.	FMT_DBLSIDE	0x0008
Drive is eight inch.	FMT_EIGHTINCH	0x0010
Drive is five inch.	FMT_FIVEINCH	0x0020
Drive is three inch.	FMT_THREEINCH	0x0040
Device is high density.	FMT_HIGHDENS	0x1000

Table 6-13. `pd_format` Available Values (Continued)

Description	Macro	EditMod
Device is standard format.	FMT_STDFMT	0x2000
Media can be removed.	FMT_REMOVABLE	0x4000
Device is a hard disk.	FMT_HARDISK	0x8000

pd_cyl
CYLNDRS

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

pd_blk
BLKSTRK

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

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

pd_sas
SEGSIZE

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

pd_ilv
INTRLV

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

pd_toffs
TRKOFFS

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

pd_boffs
BLKOFFS

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

pd_trys
TRYs

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

pd_bsize
BLKSIZE

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 6-14. pd_cntl Available Values

Description	Macro	EditMod
Disable formatting of the device	CTRL_FMTDIS	0x0
Device is capable of multi-sector transfers	CTRL_MULTI	0x1
Device size can be obtained from device	CTRL_AUTOSIZE	0x2
Device requires only one format command	CTRL_FMTENTIRE	0x3
Device needs a full track buffer for format	CTRL_TRKWRITE	0x4

pd_wpc
PRECOMP

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

pd_rwr
REDWRITE

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

pd_park
PARK

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

pd_lsnoffs
LSNOFFS

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

pd_xfersize
XFERSIZE

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 6-15. PCF Logical Unit Static Storage Fields

Field	Description File Macro
<code>v_vector</code>	VECTOR
<code>v_irqlevel</code>	IRQLEVEL
<code>v_priority</code>	PRIORITY

v_vector
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

v_irqlevel
IRQLEVEL

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

v_priority
PRIORITY

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 6-16. PCF Logical Unit Options Fields

Field	Description File Macro
lu_stp	STEP
lu_tfm	DMAMODE
lu_lun	SCSILUN
lu_ctrlrid	CTRLRID
lu_totcyls	TOTCYLS

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

Description	Macro	EditMod
30 millisecond step rate	STEP_30MS	0x00
20 millisecond step rate	STEP_20MS	0x01
12 millisecond step rate	STEP_12MS	0x02
6 millisecond step rate	STEP_6MS	0x03

lu_tfm
DMAMODE

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

lu_lun
SCSILUN

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 PCF/<DEVICE>/DESC/config.des ([Figure 6-3](#)).

Available Values

-128 to 127

lu_ctrlrid
CTRLRID

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

lu_totcyls
TOTCYLS

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



7

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.

Figure 7-1. Directory Location for Modifying Pipe Device Descriptors

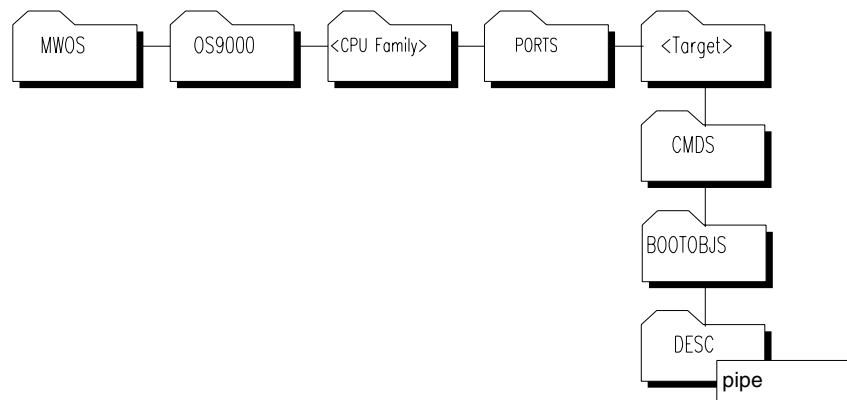
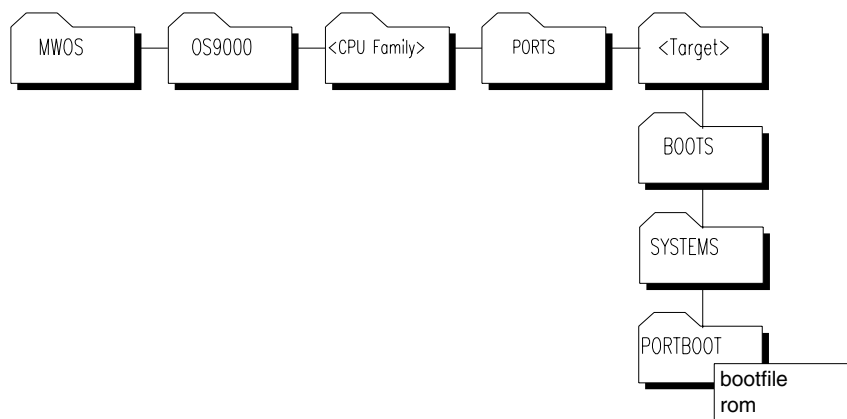


Figure 7-2. Directory Location for Modifying Low-Level Boot Images



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/BOOTOBS/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 `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 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 rom:

```
$ 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] 3
```

```
1. pipe FIFO buffer size : 0x100
```

```
$Which? [?/1-6/p/t/a/w/q] 1
```

```
pipe FIFO buffer size : 0x100
New value: 0x200
```

```
1. pipe FIFO buffer size : 0x200
```

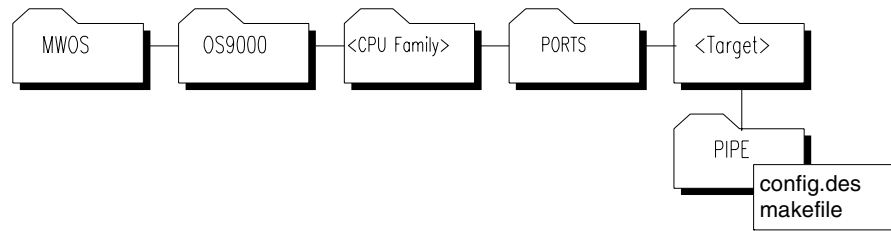
```
Which? [?/1-19/p/t/a/w/q] w
```

```
Which? [?/1-19/p/t/a/w/q] q
```

Description File Modification

You can use these procedures to modify the 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.

Figure 7-3. Directory Location for Modifying PIPE Description Files



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 7-1. Module Header Fields

Field	Description File Macro
<code>_m_group</code>	<code>MH_GROUP</code>
<code>_m_user</code>	<code>MH_USER</code>
<code>mod_name</code>	<code>MH_NAME</code>
<code>m_access</code>	<code>MH_ACCESS</code>
<code>m_tylan</code>	<code>MH_TYLAN</code>
<code>m_attrev</code>	<code>MH_ATTREV</code>
<code>m_edit</code>	<code>MH_EDITION</code>

_m_group
MH_GROUP

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

_m_user
MH_USER

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

mod_name
MH_NAME

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 `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`).

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 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 7-2. m_access Available Values

Description	Macro	EditMod
Read permission by owner	MP_OWNER_READ	0x0001
Write permission by owner	MP_OWNER_WRITE	0x0002
Execute permission by owner	MP_OWNER_EXEC	0x0004
Owner permission mask	MP_OWNER_MASK	0x000f
Read permission by group	MP_GROUP_READ	0x0010
Write permission by group	MP_GROUP_WRITE	0x0020
Execute permission by group	MP_GROUP_EXEC	0x0040

Table 7-2. `m_access` Available Values (Continued)

Description	Macro	EditMod
Group permission mask	MP_GROUP_MASK	0x00f0
Read permission by world	MP_WORLD_READ	0x0100
Write permission by world	MP_WORLD_WRITE	0x0200
Execute permission by world	MP_WORLD_EXEC	0x0400
World permission mask	MP_WORLD_MASK	0x0f00
All permissions for owner, group, and world	MP_WORLD_ACCESS	0x0777
System permission mask	MP_SYSTM_MASK	0xf000

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 `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 7-3. m_tylan Available Module Type Values

Description	Macro	EditMod
Not used (wildcard value in system calls)	MT_ANY	0x0000
Program module	MT_PROGRAM	0x0001
Subroutine module	MT_SUBROUT	0x0002
Multi-module (reserved for future use)	MT_MULTI	0x0003
Data module	MT_DATA	0x0004
Configuration data block data module	MT_CDBDATA	0x0005
Reserved for future use	0xb-0xa	0xb-0xa

Table 7-3. `m_tylan` Available Module Type Values (Continued)

Description	Macro	EditMod
User trap library	MT_TRAPLIB	0x000b
System module	MT_SYSTEM	0x000c
File manager module	MT_FILEMAN	0x000d
Physical device driver	MT_DEVDVR	0x000e
Device descriptor module	MT_DEVDESC	0x000f
User definable	0x10-0xfe	0x10-0xfe
Module type mask	MT_MASK	0xff00

Table 7-4. `m_tylan` Available Language Code Values

Description	Macro	EditMod
Unspecified language (wildcard in system calls)	ML_ANY	0x0
Machine language	ML_OBJECT	0x1
Basic I-code (reserved for future use)	ML_ICODE	0x2
Pascal P-code (reserved for future use)	ML_PCODE	0x3
C I-code (reserved for future use)	ML_CCODE	0x4
Cobol I-code (reserved for future use)	ML_CBLCODE	0x5
Fortran	ML_FRTNCODE	0x6
Reserved for future use	0x7-0xf	0x7-0xf
User-definable	0x10-0xfe	0x10-0xfe
Module language mask	ML_MASK	0x00ff

m_attrrev
MH_ATTREV

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 7-5. m_attrrev Available Attribute and Revision Values

Description	Macro	EditMod
The module is re-entrant (sharable by multiple tasks).	MA_REENT (shifted left to first byte: MA_REENT<<8)	0x80 (shifted left to first byte: 0x8000)
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.	MA_GHOST (shifted left to first byte: MA_GHOST<<8)	0x40 (shifted left to first byte: 0x4000)

Table 7-5. `m_attrrev` Available Attribute and Revision Values (Continued)

Description	Macro	EditMod
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 `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 7-6. Device Descriptor Data Definition Fields

Field	Description File Macro
<code>dd_port</code>	<code>PORTADDR</code>
<code>dd_lun</code>	<code>LUN</code>
<code>dd_pd_size</code>	<code>PD_SIZE</code>
<code>dd_type</code>	<code>DD_TYPE</code>
<code>dd_mode</code>	<code>DD_MODE</code>
<code>fmgr_name</code>	<code>FMGR_NAME</code>

Table 7-6. Device Descriptor Data Definition Fields (Continued)

Field	Description File Macro
<code>drv_name</code>	DRV_NAME
<code>dd_class</code>	DD_CLASS

dd_port
PORTADDR

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 `PIPE/config.des` ([Figure 7-3](#)).

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

```
#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

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_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 7-7. dd_type Available Values

Description	Macro	EditMod
Sequential Character File Type	DT_SCF	0x0
Random Block File Type	DT_RBF	0x1
Pipe File Type	DT_PIPE	0x2
Sequential Block File Type	DT_SBF	0x3
Network File Type	DT_NFM	0x4
Compact Disc File Type	DT_CDFM	0x5
User Communication Manager	DT_UCM	0x6
Socket Communication Manager	DT SOCK	0x7

Table 7-7. `dd_type` Available Values (Continued)

Description	Macro	EditMod
Pseudo-Keyboard Manager	DT_PTTY	0x8
Graphics File Manager	DT_GFM	0x9
PC-DOS File Manager	DT_PCF	0xa
Non-volatile RAM File Manager	DT_NRF	0xb
ISDN File Manager	DT_ISDN	0xc
MPFM File Manager	DT_MPFM	0xd
Real-Time Network File Manager	DT_RTNFM	0xe
Serial Protocol File Manager	DT_SPF	0xf
Inet File Manager	DT_INET	0xa0
Reserved for Microware Use Only	17-127	0xa1-0x7f

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_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 7-8. dd_mode Available Values for File Access Modes

Description	Macro	EditMod
Truncate on open	<code>S_ITRUNC</code>	<code>0x0100</code>
Ensure contiguous file	<code>S_ICONTIG</code>	<code>0x0400</code>
Error if file exists on create	<code>S_IEXCL</code>	<code>0x0400</code>
Create file	<code>S_ICREAT</code>	<code>0x0800</code>

Table 7-8. `dd_mode` Available Values for File Access Modes (Continued)

Description	Macro	EditMod
Append to file	<code>S_IAPPEND</code>	0x1000
Non-sharable	<code>S_ISHARE</code>	0x4000

Table 7-9. `dd_mode` Available Values for File Access Permissions

Description	Macro	EditMod
Mask for permission bits	<code>S_IPRM</code>	0xffff
Owner read	<code>S_IREAD</code>	0x0001
Owner write	<code>S_IWRITE</code>	0x0002
Owner execute	<code>S_IEXEC</code>	0x0004
Search permission	<code>S_ISEARCH</code>	0x0004
Group read	<code>S_IGREAD</code>	0x0010
Group write	<code>S_IGWRITE</code>	0x0020
Group execute	<code>S_IGEXEC</code>	0x0040
Group search	<code>S_IGSEARCH</code>	0x0040
Public read	<code>S_IOREAD</code>	0x0100
Public write	<code>S_IOWRITE</code>	0x0200
Public execute	<code>S_IOEXEC</code>	0x0400
Public search	<code>S_IOSEARCH</code>	0x0400

fmgr_name
FMGR_NAME

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`).

drv_r_name
DRV_R_NAME

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).

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 7-10. dd_class Available Values

Description	Macro	EditMod
Sequential access device	DC_SEQ	0x0001
Random access device	DC_RND	0x0002

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

Field	Description File Macro
<code>bufsz</code>	BUFSZ

bufsz
BUFSZ**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

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

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](#)
 blksiz
 changing
 for init [141](#)
 BLKSIZE
 changing [435](#) , [491](#)
 for SBF [385](#)
 BLKSTRK
 changing
 for PCF [484](#)
 for RBF [428](#)
 BLKSTRKO
 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

cha
BSPCH cha
buffer size cha
BUFBSZ cha
bufsz cha
C
C I-code mo
chd utilit
chx utilit
cinit cha
cnfgdata cha



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](#)

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_EOFFLAG [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](#)

 for SCF [175](#)
 drv_name
 changing
 for PCF [476](#)
 for pipe [532](#)
 for RBF [420](#)
 for SBF [382](#)
 for SCF [175](#)
 DSPTBLSZ
 changing
 for init [127](#)
 DT_CDFM [170](#), [377](#), [415](#), [471](#), [527](#)
 DT_GFM [171](#), [378](#), [416](#), [472](#), [528](#)
 DT_INET [171](#), [378](#), [416](#), [472](#), [528](#)
 DT_ISDN [171](#), [378](#), [416](#), [472](#), [528](#)
 DT_MPFM [171](#), [378](#), [416](#), [472](#), [528](#)
 DT_NFM [170](#), [377](#), [415](#), [471](#), [527](#)
 DT_NRF [171](#), [378](#), [416](#), [472](#), [528](#)
 DT_PCF [171](#), [378](#), [416](#), [472](#), [528](#)
 DT_PIPE [170](#), [377](#), [415](#), [471](#), [527](#)
 DT_PTTY [171](#), [378](#), [416](#), [472](#), [528](#)
 DT_RBF
 dd_type
 available values [170](#), [377](#), [415](#), [471](#), [527](#)
 DT_RTNFM [171](#), [378](#), [416](#), [472](#), [528](#)
 DT_SBF [170](#), [377](#), [415](#), [471](#), [527](#)
 DT_SCF
 dd_type
 available values [170](#), [377](#), [415](#), [471](#), [527](#)
 DT_SOCKET [170](#), [377](#), [415](#), [471](#), [527](#)
 DT_SPF [171](#), [378](#), [416](#), [472](#), [528](#)
 DT_UCM [170](#), [377](#), [415](#), [471](#), [527](#)

E

echo character
 changing [350](#)
 edition number
 of module
 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](#)
 end of file character
 changing [343](#)
 end of line character
 changing [354](#), [355](#), [356](#)
 end of record character
 changing [340](#)
 EOF character
 changing [343](#)

- for pipe [531](#)
 - for RBF [419](#)
 - for SBF [381](#)
 - for SCF [174](#)
- fmgr_name
 - changing
 - for PCF [475](#)
 - for pipe [531](#)
 - for RBF [419](#)
 - for SBF [381](#)
- for PCF
 - changing
 - PRIORITY [500](#)
- for RBF
 - changing
 - PRIORITY [444](#)
- for SBF
 - changing
 - DRIVE_FLAG [393](#)
 - PRIORITY [392](#)
- FORMAT
 - changing
 - for PCF [481](#)
 - for RBF [425](#)
- format
 - of disk
 - changing [425](#) , [481](#)
- Fortran
 - 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](#)
- FUNC0x01
 - changing
 - for SCF [217](#)
- FUNC0x02
 - changing
 - for SCF [221](#)
- FUNC0x03
 - changing
 - for SCF [225](#)
- FUNC0x04
 - changing
 - for SCF [229](#)
- FUNC0x05
 - changing
 - for SCF [233](#)
- FUNC0x06
 - changing
 - for SCF [237](#)

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
FUNCOx07															for SCF 309 , 313											
changing															FUNCOx1a											
for SCF 241															changing											
FUNCOx08															for SCF 317											
changing															FUNCOx1b											
for SCF 245															changing											
FUNCOx09															for SCF 321											
changing															FUNCOx1c											
for SCF 249															changing											
FUNCOx0a															for SCF 325											
changing															FUNCOx1d											
for SCF 253															changing											
FUNCOx0b															for SCF 329											
changing															FUNCOx1e											
for SCF 257															changing											
FUNCOx0c															for SCF 333											
changing															FUNCOx1f											
for SCF 261															changing											
FUNCOx0d															for SCF 337											
changing															FUNCOx7f											
for SCF 265															changing											
FUNCOx0e															for SCF 212											
changing																										
for SCF 269															G											
FUNCOx0f															ghost											
changing															module											
for SCF 273															setting value for cnfgdata 29											
FUNCOx10															setting value for init 101											
changing															setting value for PCF 464											
for SCF 277															setting value for pipe 520											
FUNCOx11															setting value for RBF 408											
changing															setting value for SBF 370											
for SCF 281															setting value for SCF 163											
FUNCOx12															Greenwich Mean Time (GMT) 129											
changing															group ID											
for SCF 285															module header											
FUNCOx13															cnfgdata 22											
changing															init 94											
for SCF 289															PCF 457											
FUNCOx14															pipe 513											
changing															RBF 401											
for SCF 293															SBF 363											
FUNCOx15															SCF 156											
changing															gw_address											
for SCF 297															changing											
FUNCOx16															for cnfgdata 68											
changing															interface data											
for SCF 301															configuration 68											
FUNCOx17																										
changing															H											
for SCF 305															hardware controller											
FUNCOx18															absolute physical address											
changing																										

- 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**
- 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
 - blksiz field 141
 - COMPAT 134
 - CONS_NAME 113
 - console_name field 113
 - CPUCOMPAT 128
 - desc field 144
 - dma_addr field 145
 - drive_name field 112
 - DSPTBSZ 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_tmzone field 129
 - m_tylan (type/language) field 99
 - m_user field 95
 - MAXPTY 126

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

for SCF 194

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](#)

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

- 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

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

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](#)
MT_FILEMAN [162](#)
MT_MASK [162](#)
MT_MULTI (reserved) [161](#)
MT_PROGRAM [161](#)
MT_SUBROUT [161](#)
MT_SYSTEM [162](#)
MT_TRAPLIB [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

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

- 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](#)
- maxllpmconns
 - changing
 - for cnfgdata [62](#)
- maxllpmprotos
 - 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

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

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

```
MP_WORLD_MASK
  setting module permissions
    for cnfgdata 26
    for init 98
    for PCF 461
    for pipe 517
    for RBF 405
    for SBF 367
```

- m_tylan field for init configuration data block value [99](#)
- m_tylan field for PCF configuration data block value [462](#)
- m_tylan field for pipe configuration data block value [518](#)
- m_tylan field for RBF configuration data block value [406](#)
- m_tylan field for SBF configuration data block value [368](#)
- m_tylan field for SCF configuration data block 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](#)

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

- 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
- 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](#)
 - BLKSTRKO [485](#)
 - CONTROL [492](#)
 - CTRLRID [504](#) , [505](#)
 - CYLNDRS [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](#)
 - drv_r_name field [476](#)
 - fmgr_name field [475](#)
 - FORMAT [481](#)
 - INTRLV [487](#)
 - IRQLEVEL [499](#)
 - LSNOFFS [496](#)
 - lu_ctrlrid field [504](#) , [505](#)
 - lu_lun field [503](#)

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

- 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](#)

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

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_inmap2func_code
changing
for SCF [221](#)

pd_inmap2size
changing
for SCF [222](#)

pd_inmap2string
changing
for SCF [223](#)

pd_inmap2type
changing
for SCF [220](#)

pd_inmap30func_code
changing
for SCF [333](#)

pd_inmap30size
changing

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



- 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](#)
 BLKSTRKO [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_r_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](#)

- changing [401](#)
- 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](#)

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

[drvnr_name field 175](#)
[hardware_vector field 180](#)
[m_access field 159](#)
[m_attrv \(attributes/revision\) field 163](#)
[m_edit field 165](#)
[m_group field 156](#)
[m_tylan \(type/language\) field 161](#)
[m_user field 157](#)
[pd_alf field 351](#)
[pd_backsp field 348](#)
[pd_bellch field 345](#)
[pd_bspch field 346](#)
[pd_case field 347](#)
[pd_delete field 349](#)
[pd_echo field 350](#)
[pd_eofch field 343](#)
[pd_eorch field 340](#)
[pd_inmap0func_code field 212](#)
[pd_inmap0size field 214](#)
[pd_inmap0string field 215](#)
[pd_inmap0type field 211](#)
[pd_inmap10func_code field 253](#)
[pd_inmap10size field 254](#)
[pd_inmap10string field 255](#)
[pd_inmap10type field 252](#)
[pd_inmap11func_code field 257](#)
[pd_inmap11size field 258](#)
[pd_inmap11string field 259](#)
[pd_inmap11type field 256](#)
[pd_inmap12func_code field 261](#)
[pd_inmap12size field 262](#)
[pd_inmap12string field 263](#)
[pd_inmap12type field 260](#)
[pd_inmap13func_code field 265](#)
[pd_inmap13size field 266](#)
[pd_inmap13string field 267](#)
[pd_inmap13type field 264](#)
[pd_inmap14func_code field 269](#)
[pd_inmap14size field 270](#)
[pd_inmap14string field 271](#)
[pd_inmap14type field 268](#)
[pd_inmap15func_code field 273](#)
[pd_inmap15size field 274](#)
[pd_inmap15string field 275](#)
[pd_inmap15type field 272](#)
[pd_inmap16func_code field 277](#)
[pd_inmap16size field 278](#)
[pd_inmap16string field 279](#)
[pd_inmap16type field 276](#)
[pd_inmap17func_code field 281](#)
[pd_inmap17size field 282](#)
[pd_inmap17string field 283](#)
[pd_inmap17type field 280](#)

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

- 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](#)

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

- changing [311](#)
- STRING0x19
 - changing [315](#)
- STRING0x1a
 - changing [319](#)
- STRING0x1b
 - changing [323](#)
- STRING0x1c
 - changing [327](#)
- STRING0x1d
 - changing [331](#)
- STRING0x1e
 - changing [335](#)
- STRING0x1f
 - changing [339](#)
- STRING0x7f
 - changing [215](#)
- TABCH
 - changing [344](#)
- TABSIZE
 - changing [356](#)
- TYPE0x01
 - changing [216](#)
- TYPE0x02
 - changing [220](#)
- TYPE0x03
 - changing [224](#)
- TYPE0x04
 - changing [228](#)
- TYPE0x05
 - changing [232](#)
- TYPE0x06
 - changing [236](#)
- TYPE0x07
 - changing [240](#)
- TYPE0x08
 - changing [244](#)
- TYPE0x09
 - changing [248](#)
- TYPE0x0a
 - changing [252](#)
- TYPE0x0b
 - changing [256](#)
- TYPE0x0c
 - changing [260](#)
- TYPE0x0d
 - changing [264](#)
- TYPE0x0e
 - changing [268](#)
- TYPE0x0f
 - changing [272](#)
- TYPE0x10
 - changing [276](#)

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

- 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](#)
- SIZE0x7f
 - 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 [219](#)
- STRING0x02

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

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
super user only																										
module																										
setting value for cnfgdata																										
setting value for init																										
setting value for PCF																										
setting value for pipe																										
setting value for RBF																										
setting value for SBF																										
setting value for SCF																										
SYS_DEVICE																										
changing																										
for init																										
SYS_PARAMS																										
changing																										
for init																										
SYS_PRIOR																										
changing																										
for init																										
SYS_START																										
changing																										
for init																										
SYS_TMZONE																										
changing																										
for init																										
sysboot file																										
sysgo																										
sysgo_name																										
changing																										
for init																										
system																										
time zone																										
system module																										
module header																										
type code for cnfgdata																										
type code for init																										
type code for PCF																										
type code for pipe																										
type code for RBF																										
type code for SBF																										
type code for SCF																										
system-state																										
module																										
setting value for cnfgdata																										
setting value for init																										
setting value for PCF																										
setting value for pipe																										
setting value for RBF																										
setting value for SBF																										
setting value for SCF																										
T																										
tab character																										
changing																										
TABCH																										
changing																										
for SCF																										
TABSIZE																										
changing																										
for SCF																										
Tape Device Logical Unit Number																										
changing																										
TICK_NAME																										
changing																										
for init																										
TICK_SEC																										
changing																										
for init																										
ticker_name																										
changing																										
for init																										
total number of cylinders																										
setting																										
TOTCYLS																										
changing																										
for RBF																										
track																										
number of blocks per																										
changing																										
track 0																										
number of blocks per																										
changing																										
track offset																										
changing																										
transfer memory size																										
changing																										
trap library																										
module header																										
type code for cnfgdata																										
type code for init																										
type code for PCF																										
type code for pipe																										
type code for RBF																										
type code for SBF																										
type code for SCF																										
TRKOFFS																										
changing																										
for PCF																										
for RBF																										
TRYS																										
changing																										
type																										
changing																										
for init																										
of module				</																						

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](#)

TYPE0x7f
 changing
 for SCF [211](#)

U

UPC_LOCK
 changing
 for SCF [347](#)

- for PCF 500
- for RBF 444
- for SCF 182

for SCF 182

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

wildcard
module header

language code for cnfgdata 28
language code for init 100
language code for PCF 463

```
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 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](#)

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](#)