



# **OS-9 Device Descriptor and Configuration Module Reference**

## **Version 3.1**

[www.radisys.com](http://www.radisys.com)

World Headquarters  
5445 NE Dawson Creek Drive • Hillsboro, OR  
97124 USA  
Phone: 503-615-1100 • Fax: 503-615-1121  
Toll-Free: 800-950-0044

International Headquarters  
Gebouw Flevopoort • Televisieweg 1A  
NL-1322 AC • Almere, The Netherlands  
Phone: 31 36 5365595 • Fax: 31 36 5365620

RadiSys Microwave Communications Software Division, Inc.  
1500 N.W. 118th Street  
Des Moines, Iowa 50325  
515-223-8000

Revision A  
November 2000

## Copyright and publication information

This manual reflects version 3.1 of 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.

---

November 2000  
Copyright ©2000 by RadiSys Corporation.  
All rights reserved.

EPC, INtime, iRMX, MultiPro, RadiSys, The Inside Advantage, and ValuPro are registered trademarks of RadiSys Corporation. ASM, Brahma, DAI, DAQ, MultiPro, SAIB, Spirit, and ValuePro are trademarks of RadiSys Corporation.

DAVID, MAUI, OS-9, and OS-9000, are registered trademarks of RadiSys Microware Communications Software Division, Inc. FasTrak, Hawk, SoftStax, and UpLink are trademarks of RadiSys Microware Communications Software Division, Inc.

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

---

# Table of Contents

---

## Chapter 1: Low-Level System Configuration Module (cnfgdata)7

---

8	Overview
9	cnfgdata Module Field Configuration Options
9	Direct Modification Advantages
9	Description File/Rebuild Advantages
10	Direct Modification
12	Direct Modification Procedures
13	Example EditMod Session
14	Description File Modification
14	Description File Modification Procedures
16	Low-Level Configuration Module Field Reference
17	Module Header Fields
29	Console Device Fields
46	Communication Device Fields
63	Debugger Fields
66	Low-Level Protocol Manager Fields
71	Interface Data Fields
86	Configuration Boot Data Fields
89	Boot Data Fields
96	Notification Services Field

## Chapter 2: OS-9 Configuration Module (init)99

---

100	Init Module Field Configuration Options
100	Direct Modification Advantages
100	Description File/Rebuild Advantages
101	Direct Modification
103	Direct Modification Procedures

104	Example EditMod Session
105	Description File Modification
106	Description File Modification Procedures
107	Init Module Field Reference
109	Module Header Fields
121	Module Body Fields
156	Memlist Fields
166	Cachelist Fields

### Chapter 3: SCF Device Descriptors169

---

170	SCF Field Configuration Options
170	Direct Modification Advantages
170	Description File/Rebuild Advantages
171	Direct Modification
173	Direct Modification Procedures
174	Example EditMod Session
175	Description File Modification
176	Description File Modification Procedures
177	SCF Device Descriptor Field Reference
178	Module Header Fields
190	Device Descriptor Data Definition Fields
203	SCF Description Block Fields
205	SCF Logical Unit Static Storage Fields
236	SCF Path Option Fields

### Chapter 4: SBF Device Descriptors399

---

400	SBF Field Configuration Options
400	Direct Modification Advantages
400	Description File/Rebuild Advantages
401	Direct Modification
403	Direct Modification Procedures
404	Example EditMod Session

405	Description File Configuration
405	Description File Configuration Procedures
407	SBF Device Descriptor Field Reference
408	Module Header Fields
420	Device Descriptor Data Definition Fields
432	SBF Path Options Fields
438	SBF Logical Unit Status Fields

## **Chapter 5: RBF Device Descriptors443**

---

444	RBF Field Configuration Options
444	Direct Modification Advantages
444	Description File/Rebuild Advantages
445	Direct Modification
447	Direct Modification Procedures
448	Example EditMod Session
449	Description File Configuration
449	Description File Configuration Procedures
451	RBF Device Descriptor Field Reference
452	Module Header Fields
464	Device Descriptor Data Definition Fields
476	RBF Path Option Fields
498	RBF Logical Unit Static Storage Fields
502	RBF Logical Unit Options

## **Chapter 6: PCF Device Descriptors509**

---

510	PCF Field Configuration Options
510	Direct Modification Advantages
510	Description File/Rebuild Advantages
511	Direct Modification
513	Direct Modification Procedures
514	Example EditMod Session
515	Description File Configuration

515	Description File Configuration Procedures
517	PCF Device Descriptor Field Reference
518	Module Header Fields
530	Device Descriptor Data Definition Fields
542	PCF Path Option Fields
564	PCF Logical Unit Static Storage Fields
568	PCF Logical Unit Options

## **Chapter 7: Pipe Device Descriptors**

576	Pipe Device Descriptor Field Configuration Options
576	Direct Modification Advantages
576	Description File/Rebuild Advantages
577	Direct Modification
579	Direct Modification Procedures
580	Example EditMod Session
581	Description File Modification
582	Description File Modification Procedures
583	Pipe Device Descriptor Field Reference
584	Module Header Fields
596	Device Descriptor Data Definition Fields
608	Pipeman Logical Unit Static Storage

<b>Index</b>	<b>611</b>
--------------	------------

<b>Product Discrepancy Report</b>	<b>719</b>
-----------------------------------	------------

---

# Chapter 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**



MICROWARE SOFTWARE

# 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.



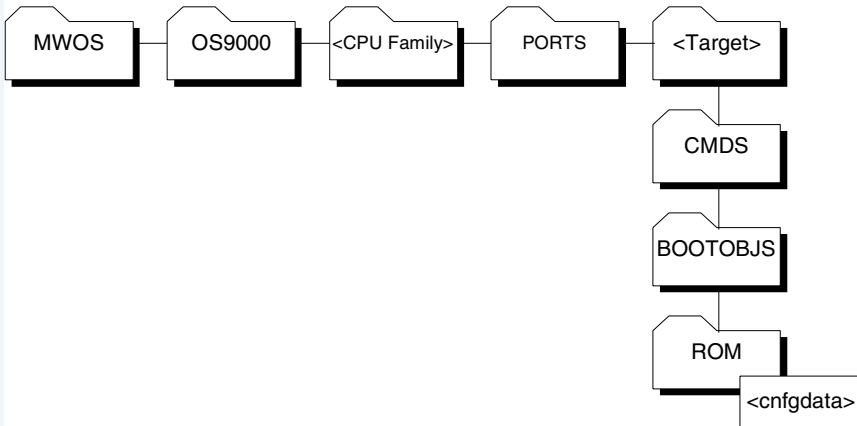
---

### For More Information

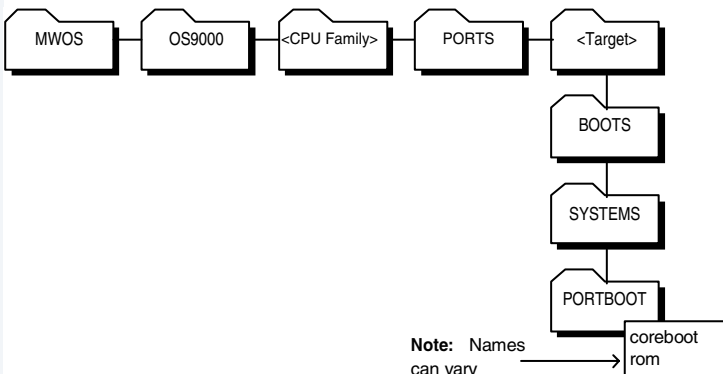
Refer to the ***Utilities Reference*** for a full description of EditMod's capabilities.

---

**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:

---

Step 1. Change to the `CMDS/BOOTOBJS/ROM` directory (see [Figure 1-1](#)).

Step 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:

---

Step 1. Change to the `BOOTS/SYSTEMS/PORTBOOT` directory (see [Figure 1-2](#)).

Step 2. Use EditMod to edit the module:

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

Step 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.

Step 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.

Step 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.

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

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

---



## Note

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)  
.
```

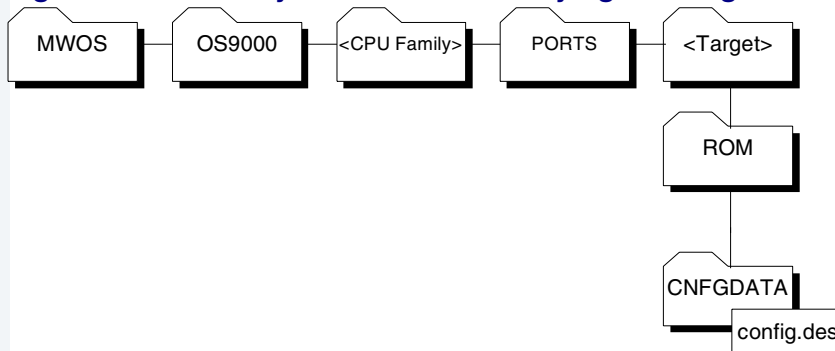
```
Which? [?/1-19/p/t/a/w/q] w
```

```
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

- Step 1. Change to the ROM/CNFGDATA directory (see [Figure 1-3](#)).
- Step 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.
- Step 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.
- Step 4. Read the comments in `config.des` to determine where to place the define for this macro.

- Step 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>
```

- Step 6. Save the changes and rebuild the module by entering the following command from the ROM/CNFGDATA directory:

```
os9make
```

- Step 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>
<code>m_attrev</code>	<code>MH_ATTREV</code>
<code>m_edit</code>	<code>MH_EDIT</code>

### **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`).

**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

**Table 1-2 m\_access Available Values (continued)**

<b>Description</b>	<b>Macro</b>	<b>EditMod</b>
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
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_SYSTEM_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 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

**Table 1-3 m\_tylan Available Module Type Values (continued)**

Description	Macro	EditMod
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
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

## 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**.



## Note

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\_attrrev Available Attribute and Revision Values**

<b>Description</b>	<b>Macro</b>	<b>EditMod</b>
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)
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 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>	<code>CONS_NAME</code>
<code>cons_vector</code>	<code>CONS_VECTOR</code>
<code>cons_priority</code>	<code>CONS_PRIORITY</code>
<code>cons_level</code>	<code>CONS_LEVEL</code>
<code>cons_timeout</code>	<code>CONS_TIMEOUT</code>
<code>cons_parity</code>	<code>CONS_PARITY</code>
<code>cons_baudrate</code>	<code>CONS_BAUDRATE</code>
<code>cons_wordsize</code>	<code>CONS_WORDSIZE</code>
<code>cons_stopbits</code>	<code>CONS_STOPBITS</code>
<code>cons_flow</code>	<code>CONS_FLOW</code>

### 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

### 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***.

### 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

**Table 1-7** cons\_parity Available Values (continued)

Description	Macro	EditMod
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

**Table 1-8** cons\_baudrate Available Values (continued)

Description	Macro	EditMod
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
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

**Table 1-8** `cons_baudrate` **Available Values (continued)**

Description	Macro	EditMod
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

**Table 1-9** `cons_wordsize` Available Values (continued)

Description	Macro	EditMod
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

**Table 1-10** `cons_stopbits` **Available Values (continued)**

Description	Macro	EditMod
Stop bit shift	CONS_STOP_SHIFT	0x40
Stop data bit shift	CONS_DBITS_SHIFT	0x60

**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>	<code>COMM_NAME</code>
<code>cons_vector</code>	<code>COMM_VECTOR</code>
<code>cons_priority</code>	<code>COMM_PRIORITY</code>
<code>cons_level</code>	<code>COMM_LEVEL</code>
<code>cons_timeout</code>	<code>COMM_TIMEOUT</code>
<code>cons_parity</code>	<code>COMM_PARITY</code>
<code>cons_baudrate</code>	<code>COMM_BAUDRATE</code>
<code>cons_wordsize</code>	<code>COMM_WORDSIZE</code>
<code>cons_stopbits</code>	<code>COMM_STOPBITS</code>
<code>cons_flow</code>	<code>COMM_FLOW</code>

**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



**EditMod Labels**

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

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

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

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

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

- This value is software dependent.

### See Also

F\_IRQ system call entry in the ***OS-9 Technical Manual***.

**EditMod Labels**

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

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

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

**See Also**

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

### **EditMod Labels**

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

### **Description**

Polling time-out value for the console device.

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

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

### **Available Values**

0 to 4294967295

**EditMod Labels**

2-configuration data  
1-console port data structure  
6-parity

**Description**

Parity mode to be used by the console device.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

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

**Table 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

**Table 1-13** `cons_parity` **Available Values (continued)**

Description	Macro	EditMod
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

**Table 1-14** `cons_baudrate` Available Values (continued)

Description	Macro	EditMod
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
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



**Table 1-14** cons\_baudrate Available Values (continued)

Description	Macro	EditMod
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

**Table 1-15** cons\_wordsize **Available Values (continued)**

Description	Macro	EditMod
Word size mask	CONS_DBITS_MASK	0xC0
Word size shift	CONS_DBITS_SHIFT	6

### 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

**Table 1-16** `cons_stopbits` **Available Values (continued)**

Description	Macro	EditMod
Stop bit shift	CONS_STOP_SHIFT	0x40
Stop data bit shift	CONS_DBITS_SHIFT	0x60

### 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>	<code>DEBUGGER_NAME</code>
<code>debug_call_at_cold</code>	<code>DEBUGGER_COLD_FLAG</code>

## **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>	<code>LLPM_MAXPROTOS</code>
<code>maxrcvmbufs</code>	<code>LLPM_MAXRCVMBUFS</code>
<code>maxllpmconns</code>	<code>LLPM_MAXCONNS</code>
<code>llpm_count</code>	<code>LLPM_COUNT</code>

**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

### **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

### **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
<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

**Table 1-21 Interface Data Fields (continued)**

Field	Description
<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.

### 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

### EditMod Labels

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

### Description

Contains the llpm interface device name string.

### Port Generic Default Value

NULL

### Port Specific Override Value

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

### Available Values

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



**port\_address****Low-level Protocol Manager Physical Address**

---

**EditMod Labels**

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

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

0 to 4294967295

### 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



### Note

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***.

**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>	<code>BOOT_COUNT</code>
<code>boot_cmdsize</code>	<code>BOOT_CMDSIZE</code>

## **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

### **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 `bootsys` 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**.



# Chapter 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**



MICROWARE SOFTWARE

# 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.



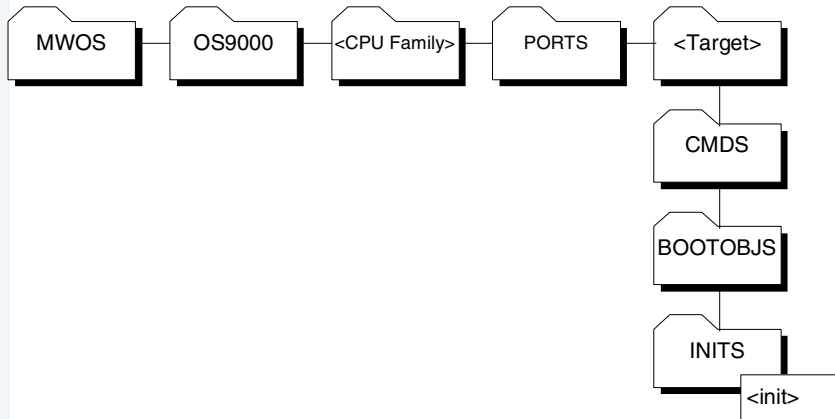
---

### For More Information

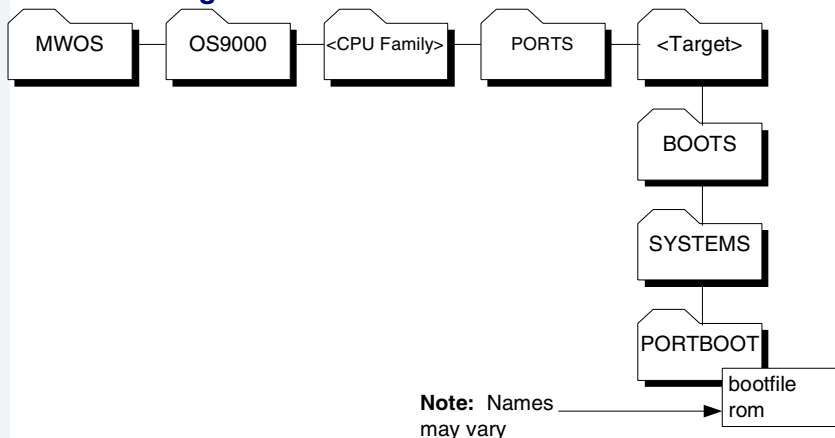
Refer to the ***Utilities Reference*** for a full description of `EditMod`'s capabilities.

---

**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**



## For More Information

See 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:

---

Step 1. Change to the `CMDS/BOOTOBJS/INITS` directory (see [Figure 2-1](#)).

Step 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:

---

Step 1. Change to the `BOOTS/SYSTEMS/PORTBOOT` directory (see [Figure 2-2](#)).

Step 2. Use `EditMod` to edit the module:

```
$EditMod -e init -f=<boot image name>
```

Step 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.

Step 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.

Step 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.

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

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

---



## Note

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]
```

```
.
.
.
.
```

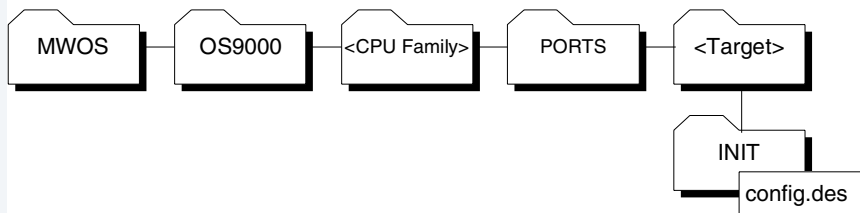
```
Which? [?/1-19/p/t/a/w/q] w
```

```
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

---

- Step 1. Change to the `INIT` directory. (see [Figure 2-3](#)).
- Step 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.
- Step 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.
- Step 4. Read the comments in `config.des` to determine where to place the define for this macro.
- Step 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>
```
- Step 6. Save the changes and rebuild the module, entering the following command in the `INIT` directory:
- ```
os9make
```
- Step 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.



---

### Note

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>

### **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

### 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`).



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

**Table 2-2** `m_access` Available Values (continued)

Description	Macro	EditMod
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
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_SYSTEM_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 `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_MULTII	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
System module	MT_SYSTEM	0x000c
File manager module	MT_FILEMAN	0x000d
Physical device driver	MT_DEVDVR	0x000e
Device descriptor module	MT_DEVDESC	0x000f

**Table 2-3** `m_tylan` Available Module Type Values (continued)

Description	Macro	EditMod
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

## 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](#).



## Note

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**

<b>Description</b>	<b>Macro</b>	<b>EditMod</b>
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)
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. 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
<code>m_site</code>	<code>SITE</code>
<code>m_cputyp</code>	<code>MPUCHIP</code>
<code>install_name</code>	<code>INSTALNAME</code>
<code>os9rev_name</code>	<code>OS9K_REVSTR</code>
<code>sysgo_name</code>	<code>SYS_START</code>
<code>sparam_string</code>	<code>SYS_PARAMS</code>
<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>

**Table 2-6 Module Header Fields (continued)**

Field	Description File Macro
<code>m_procs</code>	PROCS
<code>m_paths</code>	PATHS
<code>m_events</code>	EVENTS
<code>m_ticksec</code>	TICK_SEC
<code>m_slice</code>	SLICE
<code>m_syspri</code>	SYS_PRIOR
<code>m_minpty</code>	MINPTY
<code>m_maxage</code>	MAXPTY
<code>m_dsptbl</code>	DSPTBLSZ
<code>m_cpucompat</code>	CPUCOMPAT
<code>m_tmzone</code>	SYS_TMZONE
<code>m_level</code>	OS_LEVEL
<code>m_major</code>	OS_VERSION
<code>m_minor</code>	OS_REVISION
<code>m_edition</code>	OS_EDITION
<code>m_compat</code>	COMPAT

**Table 2-6 Module Header Fields (continued)**

Field	Description File Macro
m_maxsig	MAXSIG
preio_name	PREIOS_NAME

### 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`).

### **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`).

### 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`).



**EditMod Labels**

2-init module contents

15-number of process table entries

**Description**

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

**Port Generic Default Value**

64 (0x40)

**Port Specific Override Value**

Refer to `INIT/config.des` ([Figure 2-3](#)).

**Available Values**

0 to 65535

**EditMod Labels**

2-init module contents  
16-number of path table entries

**Description**

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

**Port Generic Default Value**

64 (0x40)

**Port Specific Override Value**

Refer to `INIT/config.des` ([Figure 2-3](#)).

**Available Values**

0 to 65535

### EditMod Labels

2-init module contents

17-number of system event table entries

### Description

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



---

### For More Information

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

---

### Port Generic Default Value

32 (0x20)

### Port Specific Override Value

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

### Available Values

0 to 65535

### EditMod Labels

2-init module contents  
18-number of ticks per second

### Description

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

### Port Generic Default Value

0 (zero)

### Port Specific Override Value

Refer to `INIT/config.des` ([Figure 2-3](#)).

### Available Values

0 to 65535

**EditMod Labels**

2-init module contents

19-number of clock ticks per time slice

**Description**

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

**Port Generic Default Value**

2

**Port Specific Override Value**

Refer to `INIT/config.des` ([Figure 2-3](#)).

**Available Values**

0 to 65535

### EditMod Labels

2-init module contents  
20-initial system priority

### Description

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

### Port Generic Default Value

128 (0x80)

### Port Specific Override Value

Refer to `INIT/config.des` ([Figure 2-3](#)).

### Available Values

0 to 65535

**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

### 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

**EditMod Labels**

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

**Description**

This field is reserved for system-specific flags.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to `INIT/config.des` ([Figure 2-3](#)).

**Available Values**

0 to 65535

**m\_tmzone**SYS\_TMZONE

---

**EditMod Labels**

2-init module contents

25-system time zone

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to `INIT/config.des` ([Figure 2-3](#)).

**Available Values**

-32768 to 32767

### **EditMod Labels**

2-init module contents  
26-OS-9000 level

### **Description**

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

### **Port Generic Default Value**

1

### **Port Specific Override Value**

Refer to `INIT/config.des` ([Figure 2-3](#)).

### **Available Values**

0 to 255

**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

**EditMod Labels**

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

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to `INIT/config.des` ([Figure 2-3](#)).

**Available Values**

0 to 255

**EditMod Labels**

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

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to `INIT/config.des` ([Figure 2-3](#)).

**Available Values**

0 to 255

EditMod Labels

2-init module contents  
30-compatibility flags

Description

This byte is used for revision compatibility.

Port Generic Default Value

Macro

B\_WIPEMEM

EditMod

0x2

Port Specific Override Value

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

Available Values

Revision compatibility values are located in the header file, module.h., and are listed in Table 2-7.

Table 2-7 m\_compat Available Attribute and Revision Values

Description	Macro	EditMod
Set to ignore sticky bit in the module headers	B_GHOST	0x01
Set to patternize memory when allocated and returned	B_WIPEMEM	0x02



**Table 2-7** `m_compat` **Available Attribute and Revision Values (continued)**

Description	Macro	EditMod
Set to inform the kernel not to automatically set the clock during coldstart	B_NOCLOCK	0x04
Set to not automatically expand system tables	B_EXPTBL	0x08
Set to not use SSM	B_SSM	0x16

### 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
<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

### 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

**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](#).



---

**Note**

Only `B_USERRAM` memory can be initialized.

---

**Table 2-10 Access Bit Definitions for Memory Type**

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



**EditMod Labels**

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

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

0 to 65535

**EditMod Labels**

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

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to `INIT/config.des` ([Figure 2-3](#)).

**Available Values**

0 to 4294967295

**EditMod Labels**

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

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to `INIT/config.des` ([Figure 2-3](#)).

**Available Values**

0 to 4294967295

### 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



---

# Chapter 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**



MICROWARE SOFTWARE

# 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.



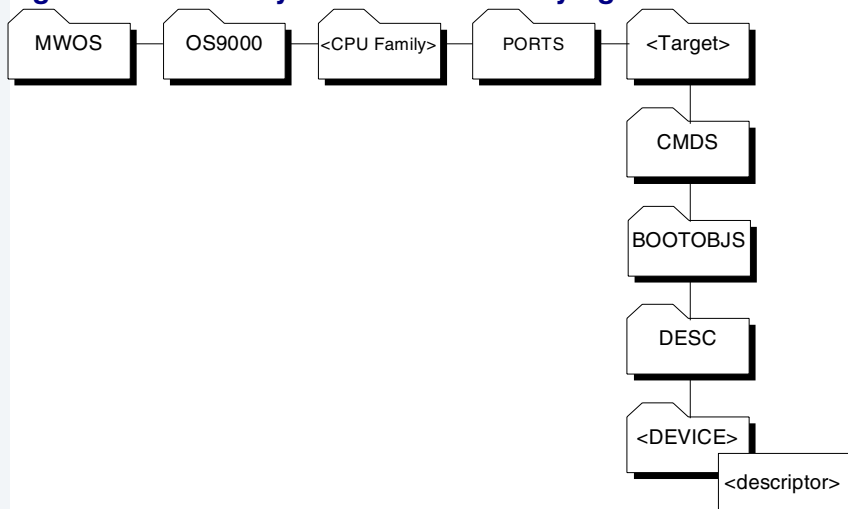
---

### For More Information

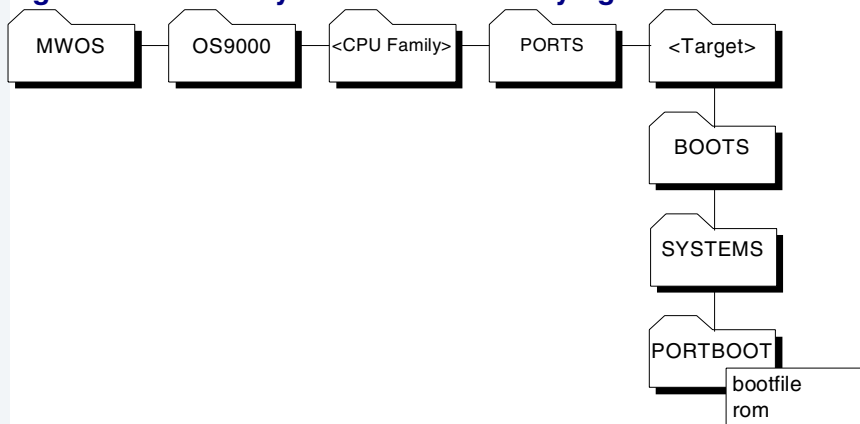
Refer to the ***Utilities Reference*** for a full description of `EditMod`'s capabilities.

---

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



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



## For More Information

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:

---

Step 1. Go to the `CMDS/BOOTOBJS/DESC/<DEVICE>` directory (see [Figure 3-1](#)).

Step 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:

---

Step 1. Go to the `BOOTS/SYSTEMS/PORTBOOT` directory (see [Figure 3-2](#)).

Step 2. Use `EditMod` to edit the module:

```
$EditMod -e <descriptor> -f=<boot image name>
```

Step 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.

Step 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.

Step 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.

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

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

---



## Note

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
```

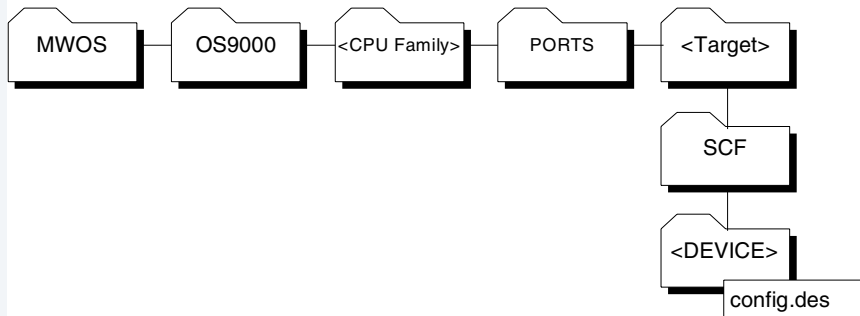
```
Which? [?/1-19/p/t/a/w/q] w
```

```
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

- 
- Step 1. Change to the `SCF / <DEVICE>` directory (see [Figure 3-3](#)).
  - Step 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.
  - Step 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.
  - Step 4. Read the comments in `config.des` to determine where to place the define for this macro.
  - Step 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>
```
  - Step 6. Save the changes and rebuild the SCF device descriptors, entering the following command in the `SCF / <DEVICE> / DESC` directory:  
  

```
os9make
```
  - Step 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>
<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

### **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).

## 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

**Table 3-2** `m_access` Available Values (continued)

Description	Macro	EditMod
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
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 `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
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

**Table 3-3 m\_tylan Available Module Type Values (continued)**

Description	Macro	EditMod
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**.

**Note**

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\_attr Available Attribute and Revision Values**

<b>Description</b>	<b>Macro</b>	<b>EditMod</b>
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)
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 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>	<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>
<code>drvr_name</code>	<code>DRVR_NAME</code>
<code>dd_class</code>	<code>DD_CLASS</code>

**EditMod Labels**

2-device descriptor data definitions  
1-device port address

**Description**

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

**Macro Example**

```
#define PORTADDR      0xfffe4000
```

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

0 to 4294967295

### **EditMod Labels**

2-device descriptor data definitions  
2-logical unit number

### **Description**

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

### **Macro Example**

```
#define LUN      2
```

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

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

### **Available Values**

0 to 65535



**dd\_pd\_size**PD\_SIZE

---

**EditMod Labels**

2-device descriptor data definitions  
3-path descriptor size

**Description**

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

**Port Generic Default Value**

0x234

**Port Specific Override Value**

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

**Available Values**

0 to 65535

EditMod Labels

2-device descriptor data definitions  
4-device type

Description

Identifies the I/O class of the device.

Port Generic Default Value

Macro

DT\_SCF

EditMod

0x0

Port Specific Override Value

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

Available Values

Device type values are defined in the header file `io.h`, and are listed in [Table 3-7](#).

Table 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

**Table 3-7** `dd_type` Available Values (continued)

Description	Macro	EditMod
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
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

**Table 3-7** `dd_type` Available Values (continued)

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

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

Description	Macro	EditMod
Mask for permission bits	<code>S_IPRM</code>	<code>0xffff</code>
Owner read	<code>S_IREAD</code>	<code>0x0001</code>
Owner write	<code>S_IWRITE</code>	<code>0x0002</code>
Owner execute	<code>S_IEXEC</code>	<code>0x0004</code>
Search permission	<code>S_ISEARCH</code>	<code>0x0004</code>

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

Description	Macro	EditMod
Group read	<code>S_IGREAD</code>	<code>0x0010</code>
Group write	<code>S_IGWRITE</code>	<code>0x0020</code>
Group execute	<code>S_IGEXEC</code>	<code>0x0040</code>
Group search	<code>S_IGSEARCH</code>	<code>0x0040</code>
Public read	<code>S_IOREAD</code>	<code>0x0100</code>
Public write	<code>S_IOWRITE</code>	<code>0x0200</code>
Public execute	<code>S_IOEXEC</code>	<code>0x0400</code>
Public search	<code>S_IOSEARCH</code>	<code>0x0400</code>

**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).

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
<code>v_irqmask</code>	IRQ_MASK
<code>v_maxbuff</code>	MAXBUFF
<code>v_insize</code>	INSIZE
<code>v_outsize</code>	OUTSIZE
<code>v_line</code>	PAGE_SIZE
<code>v_intr</code>	KYBDINTR
<code>v_quit</code>	KYBDQUIT

**Table 3-12 Device Descriptor Data Definition Fields (continued)**

Field	Description File Macro
<code>v_psch</code>	KYBDPAUSE
<code>v_xon</code>	XON
<code>v_xoff</code>	XOFF
<code>v_baud</code>	BAUDRATE
<code>v_parity</code>	LUPARITY
<code>v_stopbits</code>	STOPBITS
<code>v_wordsize</code>	WORDSIZE
<code>v_rtsstate</code>	RTSSTATE
<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

**EditMod Labels**

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

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

0 to 65535

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



**EditMod Labels**

4-SCF logical unit static storage  
3-irq polling priority

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

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

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

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

## See Also

F\_IRQ system call entry in the ***OS-9 Technical Manual***.

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

### EditMod Labels

4-SCF logical unit static storage  
5-pollled 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

**EditMod Labels**

4-SCF logical unit static storage

6-driver accessible copy of logical unit number

**Description**

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

**Macro Example**

```
#define LUN 2
```

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

0 to 65535

### EditMod Labels

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

### Description

This is the interrupt mask for the SCF device.

### Port Generic Default Value

0 (zero)

### Port Specific Override Value

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

### Available Values

0 to 65535

**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

### **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

### **EditMod Labels**

4-SCF logical unit static storage  
13-lines left until end of page

### **Description**

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

### **Port Generic Default Value**

24

### **Port Specific Override Value**

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

### **Available Values**

0 to 65535

EditMod Labels

4-SCF logical unit static storage  
14-keyboard interrupt character

Description

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

Port Generic Default Value

Macro

CTRL\_C

EditMod

0x03

Port Specific Override Value

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

Available Values

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

Table 3-15 ASCII Control Character Available Values

SCF/OS-9 Compatible Standard Codes	Macro	EditMod
	C_NULL	0x00
C_REPEAT	CTRL_A	0x01

**Table 3-15 ASCII Control Character Available Values (continued)**

<b>SCF/OS-9 Compatible Standard Codes</b>	<b>Macro</b>	<b>EditMod</b>
	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
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

**Table 3-15 ASCII Control Character Available Values (continued)**

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

## 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**.

## 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**.

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

**Table 3-16** cons\_baudrate Available Values (continued)

Description	Macro	EditMod
110 bps	BAUD110	0x03
134.5 bps	BAUD134P5	0x04
150 bps	BAUD150	0x05
300 bps	BAUD300	0x06
600 pbs	BAUD600	0x07
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

**Table 3-16** `cons_baudrate` Available Values (continued)

Description	Macro	EditMod
57,600 bps	BAUD57600	0x14
64,000 bps	BAUD64000	0x15
115,200 bps	BAUD115200	0x16

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

**Table 3-17** `v_parity` Available Values (continued)

Description	Macro	EditMod
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

## 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\_wordsizewordsize Available Values**

Description	Macro	EditMod
5 bits per character	WORDSIZE5	0x5
6 bits per character	WORDSIZE6	0x6



**Table 3-19** `v_wordsize` **Available Values (continued)**

Description	Macro	EditMod
7 bits per character	WORDSIZE7	0x7
8 bits per character	WORDSIZE8	0x8

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	RTSEENABLED	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
<code>pd_inmap0type</code>	<code>TYPE0x7f</code>
<code>pd_inmap0func_code</code>	<code>FUNC0x7f</code>
<code>pd_inmap0size</code>	<code>SIZE0x7f</code>
<code>pd_inmap0string</code>	<code>STRING0x7f</code>
<code>pd_inmap1type</code>	<code>TYPE0x01</code>
<code>pd_inmap1func_code</code>	<code>FUNC0x01</code>
<code>pd_inmap1size</code>	<code>SIZE0x01</code>
<code>pd_inmap1string</code>	<code>STRING0x01</code>
<code>pd_inmap2type</code>	<code>TYPE0x02</code>
<code>pd_inmap2func_code</code>	<code>FUNC0x02</code>
<code>pd_inmap2size</code>	<code>SIZE0x02</code>
<code>pd_inmap2string</code>	<code>STRING0x02</code>
<code>pd_inmap3type</code>	<code>TYPE0x03</code>

**Table 3-21 Device Descriptor Data Definition Fields (continued)**

Field	Description File Macro
<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>

**Table 3-21 Device Descriptor Data Definition Fields (continued)**

Field	Description File Macro
<code>pd_inmap7size</code>	<code>SIZE0x07</code>
<code>pd_inmap7string</code>	<code>STRING0x07</code>
<code>pd_inmap8type</code>	<code>TYPE0x08</code>
<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>

**Table 3-21 Device Descriptor Data Definition Fields (continued)**

Field	Description File Macro
<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>
<code>pd_inmap13string</code>	<code>STRING0x0d</code>
<code>pd_inmap14type</code>	<code>TYPE0x0e</code>
<code>pd_inmap14func_code</code>	<code>FUNC0x0e</code>
<code>pd_inmap14size</code>	<code>SIZE0x0e</code>
<code>pd_inmap14string</code>	<code>STRING0x0e</code>
<code>pd_inmap15type</code>	<code>TYPE0x0f</code>
<code>pd_inmap15func_code</code>	<code>FUNC0x0f</code>
<code>pd_inmap15size</code>	<code>SIZE0x0f</code>
<code>pd_inmap15string</code>	<code>STRING0x0f</code>

**Table 3-21 Device Descriptor Data Definition Fields (continued)**

Field	Description File Macro
<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>
<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>



**Table 3-21 Device Descriptor Data Definition Fields (continued)**

<b>Field</b>	<b>Description File Macro</b>
<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>

**Table 3-21 Device Descriptor Data Definition Fields (continued)**

Field	Description File Macro
<code>pd_inmap24size</code>	<code>SIZE0x18</code>
<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>

**Table 3-21 Device Descriptor Data Definition Fields (continued)**

Field	Description File Macro
<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>
<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>

**Table 3-21 Device Descriptor Data Definition Fields (continued)**

Field	Description File Macro
<code>pd_bspch</code>	BSPCH
<code>pd_case</code>	UPC_LOCK
<code>pd_backsp</code>	BSB
<code>pd_delete</code>	LINEDEL
<code>pd_echo</code>	AUTOECHO
<code>pd_alf</code>	AUTOLF
<code>pd_pause</code>	PAGEPAUSE
<code>pd_insm</code>	INSERTMODE
<code>pd_nulls</code>	EOLNULLS
<code>pd_page</code>	PAGESIZE
<code>pd_tabsiz</code>	TABSIZE

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

**Table 3-22 ASCII Control Character Available Values (continued)**

Control Character is...	Macro	EditMod
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

**Table 3-23 ASCII Control Character Available Values (continued)**

<b>Description</b>	<b>Macro</b>	<b>EditMod</b>
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
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	DELINEL	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**.

### **EditMod Labels**

5-SCF path options  
6-'\x01' editing function code

### **Description**

SCF editing function mapping code for specified character.

### **Port Generic Default Value**

Macro

MOVEND

EditMod

0x07

### **Port Specific Override Value**

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

### **Available Values**

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

**pd\_inmap1size**SIZE0x01

---

**EditMod Labels**

5-SCF path options

7-'\x01' size of associated string

**Description**

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

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

0 to 4294967295

## 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**.

### **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**.

## 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**.

## 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**.

### 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**.

## 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\_inmap1size**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**.

### 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**.

### 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**.

## 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**.

### **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

DELIN

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

**Table 3-24 ASCII Control Character Available Values (continued)**

<b>SCF/OS-9 Compatible Standard Codes</b>	<b>Macro</b>	<b>EditMod</b>
	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
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



**Table 3-24 ASCII Control Character Available Values (continued)**

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

## EditMod Labels

5-SCF path options  
130-end of file character

## Description

This specifies the end of file character.

## Port Generic Default Value

Macro

EOFCH (defined as C\_EOF in `scfdesc.h`)

EditMod

0x1B

## Port Specific Override Value

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

## Available Values

The ASCII control and special characters are defined in the header file, `scf.h`, and in **Table 3-24**.

**EditMod Labels**

5-SCF path options  
131-tab character (0 = none)

**Description**

This defines the tab character.

**Port Generic Default Value**

Macro

TABCH (defined as C\_TAB in `scfdesc.h`)

EditMod

0x09

**Port Specific Override Value**

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

**Available Values**

The ASCII control and special characters are defined in the header file, `scf.h`, and in **Table 3-24**.

## EditMod Labels

5-SCF path options  
132-bell (line overflow)

## Description

This defines the bell character.

## Port Generic Default Value

Macro

BELLCH (defined as C\_BELL in scfdesc.h)

EditMod

0x07

## Port Specific Override Value

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

## Available Values

The ASCII control and special characters are defined in the header file, `scf.h`, and in **Table 3-24**.

**EditMod Labels**

5-SCF path options  
133-backspace echo character

**Description**

This defines the backspace echo character.

**Port Generic Default Value**

Macro

BSPCH (defined as C\_BACKSPACE in `scfdesc.h`)

EditMod

`'\b'`

**Port Specific Override Value**

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

**Available Values**

The ASCII control and special characters are defined in the header file, `scf.h`, and in **Table 3-24**.

## EditMod Labels

5-SCF path options  
134-case lock

## Description

This specifies the state of the upper case lock character.

## Port Generic Default Value

The default is upper and lower case.

Macro

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

## 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

## 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

## 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	PLOFF	0x00
Positive logic on - Echo on	PLON	0x01
Negative logic off	NLOFF	0x00
Negative logic on	NLON	0x01

## 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

## 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

## 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

### **EditMod Labels**

5-SCF path options  
141-end of line null count

### **Description**

This specifies the number of end of line nulls.

### **Port Generic Default Value**

0 (no end of line nulls)

### **Port Specific Override Value**

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

### **Available Values**

-128 to 127

**EditMod Labels**

5-SCF path options

142-lines per page

**Description**

This specifies the number of lines per page.

**Port Generic Default Value**

24

**Port Specific Override Value**

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

**Available Values**

-128 to 127

### **EditMod Labels**

5-SCF path options  
143-tab field size

### **Description**

This specifies the number of spaces a tab skips.

### **Port Generic Default Value**

4

### **Port Specific Override Value**

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

### **Available Values**

-128 to 127

---

# Chapter 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**



MICROWARE SOFTWARE

# 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

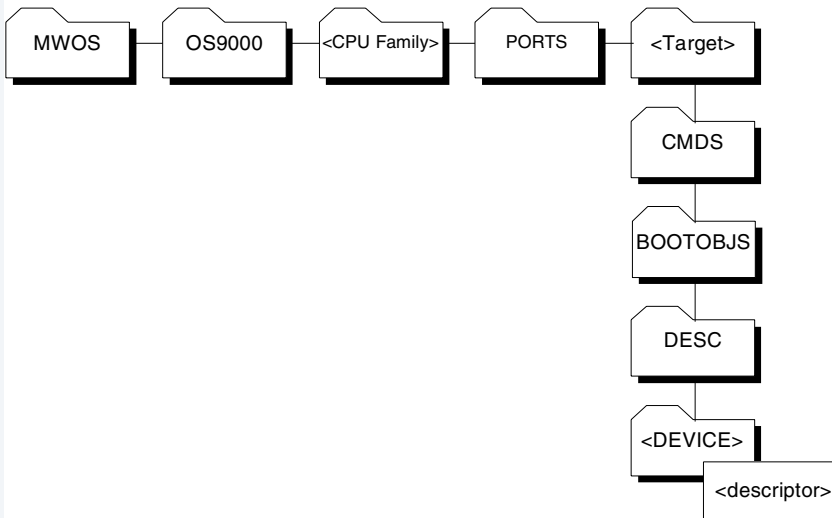
Use the `Editmod` utility and the following procedures to directly modify fields in the existing SBF device descriptor module. The module can stand-alone or it can be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The 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.



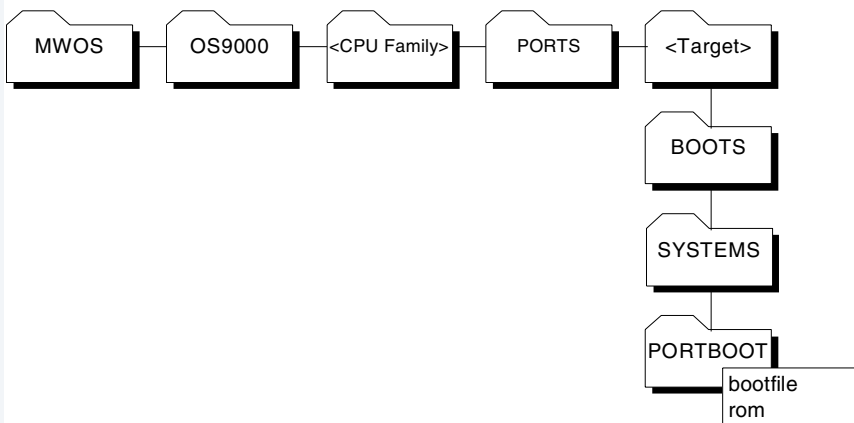
## For More Information

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**





---

## For More Information

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:

---

Step 1. Change to the `CMD5/BOOTBJS/DESC/<DEVICE>` directory (see [Figure 4-1](#)).

Step 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:

---

Step 1. Change to the `BOOTS/SYSTEMS/PORTBOOT` directory (see [Figure 4-2](#)).

Step 2. Use EditMod to edit the module:

```
$EditMod -e <descriptor> -f=<boot image name>
```

Step 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.

Step 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.

- Step 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.
- Step 6. Select the `w` command (write) to save the changes.
- Step 7. Select the `q` command (quit) to exit EditMod.
- 



## Note

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
```

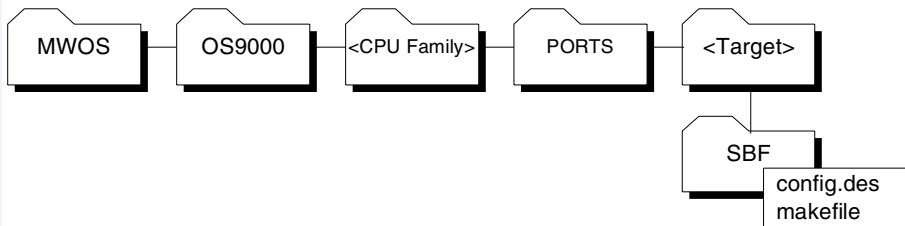
Which? [?/1-19/p/t/a/w/q] **w**

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

- Step 1. Change to the SBF/<DEVICE> directory (see [Figure 4-3](#)).
- Step 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.
- Step 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.

- Step 4. Read the comments in `config.des` to determine where to place the define for this macro.
- Step 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>
```
- Step 6. Save the changes and rebuild the SBF device descriptors, entering the following command in the `SBF/<DEVICE>/DESC` directory:
- ```
os9make
```
- Step 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> | <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 SBF/<DEVICE>/DESC/config.des ([Figure 4-3](#)).

**Available Values**

0 to 65535

### **EditMod Labels**

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

### **Description**

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

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

Refer to SBF/<DEVICE>/DESC/config.des (**Figure 4-3**).

### **Available Values**

0 to 65535

**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).

## 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  |

**Table 4-2** `m_access` Available Values (continued)

| Description                                 | Macro           | EditMod |
|---------------------------------------------|-----------------|---------|
| 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  |
| 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_SYSTEM_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 |
| 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  |

**Table 4-3 m\_tylan Available Module Type Values (continued)**

| Description      | Macro     | EditMod   |
|------------------|-----------|-----------|
| 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**.



### Note

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\_attrrev Available Attribute and Revision Values**

| <b>Description</b>                                                                                                                      | <b>Macro</b>                                             | <b>EditMod</b>                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|-------------------------------------------------|
| The module is re-entrant (sharable by multiple tasks).                                                                                  | MA_REENT<br>(shifted left to first byte:<br>MA_REENT<<8) | 0x80<br>(shifted left to first byte:<br>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<br>(shifted left to first byte:<br>MA_GHOST<<8) | 0x40<br>(shifted left to first byte:<br>0x4000) |
| The module is a system-state module.                                                                                                    | MA_SUPER<br>(shifted left to first byte:<br>MA_SUPER<<8) | 0x20<br>(shifted left to first byte:<br>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 |
|-------------------------|------------------------|
| <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>dd_port</code>    | <code>MFGR_NAME</code> |
| <code>drvr_name</code>  | <code>DRVR_NAME</code> |
| <code>dd_class</code>   | <code>DD_CLASS</code>  |

**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

### **EditMod Labels**

2-device descriptor data definitions  
2-logical unit number

### **Description**

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

### **Macro Example**

```
#define LUN      2
```

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

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

### **Available Values**

0 to 65535

**dd\_pd\_size**PD\_SIZE

---

**EditMod Labels**

2-device descriptor data definitions  
3-path descriptor size

**Description**

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

**Port Generic Default Value**

124

**Port Specific Override Value**

Refer to SBF/<DEVICE>/DESC/config.des (**Figure 4-3**).

**Available Values**

0 to 65535

## EditMod Labels

2-device descriptor data definitions  
4-device type

## Description

Identifies the I/O class of the device.

## Port Generic Default Value

Macro

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     |



**Table 4-7** `dd_type` Available Values (continued)

| Description                     | Macro    | EditMod   |
|---------------------------------|----------|-----------|
| 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       |
| 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 |

## 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> |
| Append to file                 | <code>S_IAPPEND</code> | <code>0x1000</code> |
| Non-sharable                   | <code>S_ISHARE</code>  | <code>0x4000</code> |

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

| Description              | Macro                  | EditMod             |
|--------------------------|------------------------|---------------------|
| Mask for permission bits | <code>S_IPRM</code>    | <code>0xffff</code> |
| Owner read               | <code>S_IREAD</code>   | <code>0x0001</code> |
| Owner write              | <code>S_IWRITE</code>  | <code>0x0002</code> |
| Owner execute            | <code>S_IEXEC</code>   | <code>0x0004</code> |
| Search permission        | <code>S_ISEARCH</code> | <code>0x0004</code> |

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

| Description    | Macro                   | EditMod             |
|----------------|-------------------------|---------------------|
| Group read     | <code>S_IGREAD</code>   | <code>0x0010</code> |
| Group write    | <code>S_IGWRITE</code>  | <code>0x0020</code> |
| Group execute  | <code>S_IGEXEC</code>   | <code>0x0040</code> |
| Group search   | <code>S_IGSEARCH</code> | <code>0x0040</code> |
| Public read    | <code>S_IOREAD</code>   | <code>0x0100</code> |
| Public write   | <code>S_IOWRITE</code>  | <code>0x0200</code> |
| Public execute | <code>S_IOEXEC</code>   | <code>0x0400</code> |
| Public search  | <code>S_IOSEARCH</code> | <code>0x0400</code> |

**fmgr\_name**FMGR\_NAME

---

**EditMod Labels**

2-device descriptor data definitions

6-file manager name

**Description**

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

**Port Generic Default Value**`"sbf"`**Port Specific Override Value**

Refer to SBF/<DEVICE>/DESC/config.des (**Figure 4-3**).

**Available Values**

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

### **EditMod Labels**

2-device descriptor data definitions  
7-driver name

### **Description**

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

### **Port Generic Default Value**

NULL

### **Port Specific Override Value**

Refer to SBF/<DEVICE>/DESC/config.des (**Figure 4-3**).

### **Available Values**

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

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\_blksiz**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 compatability flags

### Description

SBF driver compatibility flags.

### Port Generic Default Value

0 (zero)

### Port Specific Override Value

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

### Available Values

Compatibility flag values are defined in the header file `sbf.h`, and in [Table 4-12](#).

**Table 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

### **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

### 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

### EditMod Labels

4-SBF logical unit status  
4-drive flag

### Description

Current state of SBF device.

### Port Generic Default Value

0 (zero)

### Port Specific Override Value

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

### Available Values

Drive flag values are defined in the header file `sbf.h`, and in [Table 4-14](#).

**Table 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  |

---

# Chapter 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**



MICROWARE SOFTWARE

# 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

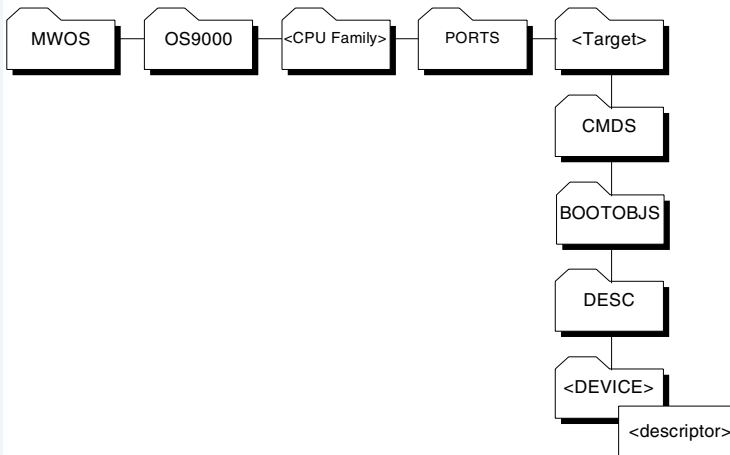
Use the `Editmod` utility and the following procedures to directly modify fields in the existing RBF device descriptor module. The module can stand-alone or it can be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The 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.



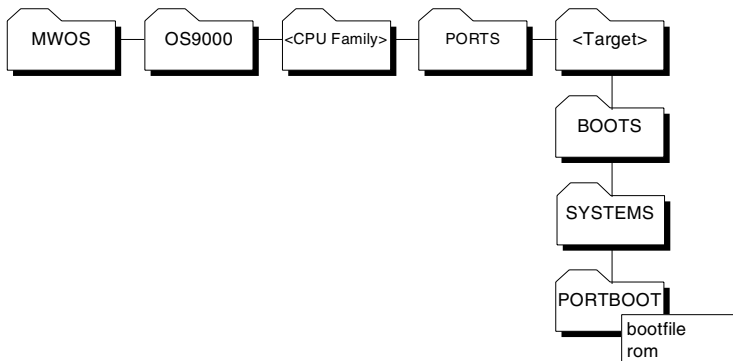
## For More Information

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**






---

## For More Information

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:

---

Step 1. Change to the `CMDS/BOOTBJS/DESC/<DEVICE>` directory (see **Figure 5-1**).

Step 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:

---

Step 1. Change to the `BOOTS/SYSTEMS/PORTBOOT` directory (see **Figure 5-2**).

Step 2. Use EditMod to edit the module:

```
$EditMod -e <descriptor> -f=<boot image name>
```

Step 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.

Step 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.

- Step 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.
- Step 6. Select the `w` command (write) to save the changes.
- Step 7. Select the `q` command (quit) to exit EditMod.
- 



## Note

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
```



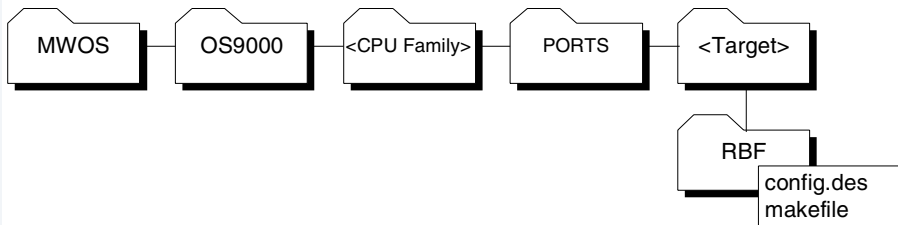
Which? [?/1-19/p/t/a/w/q] **w**

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

- Step 1. Change to the RBF/<DEVICE> directory (see **Figure 5-3**).
- Step 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.
- Step 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.

- Step 4. Read the comments in `config.des` to determine where to place the define for this macro.
- Step 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>
```
- Step 6. Save the changes and rebuild the RBF device descriptors, entering the following command in the `RBF/<DEVICE>/DESC` directory:
- ```
os9make
```
- Step 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>	<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 RBF/<DEVICE>/DESC/config.des ([Figure 5-3](#)).

**Available Values**

0 to 65535

### **EditMod Labels**

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

### **Description**

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

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

### **Available Values**

0 to 65535

**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`).

## 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



**Table 5-2** `m_access` Available Values (continued)

Description	Macro	EditMod
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
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_SYSTEM_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 `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
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

**Table 5-3** `m_tylan` Available Module Type Values (continued)

Description	Macro	EditMod
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\_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 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**.



---

**Note**

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\_attrrev Available Attribute and Revision Values**

<b>Description</b>	<b>Macro</b>	<b>EditMod</b>
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)
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
<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>dd_port</code>	<code>MFGR_NAME</code>
<code>drvr_name</code>	<code>DRVR_NAME</code>
<code>dd_class</code>	<code>DD_CLASS</code>



**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

### **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

## 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

**Table 5-7** `dd_type` Available Values (continued)

Description	Macro	EditMod
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
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

## 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>
Append to file	<code>S_IAPPEND</code>	<code>0x1000</code>
Non-sharable	<code>S_ISHARE</code>	<code>0x4000</code>

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

Description	Macro	EditMod
Mask for permission bits	<code>S_IPRM</code>	<code>0xffff</code>
Owner read	<code>S_IREAD</code>	<code>0x0001</code>
Owner write	<code>S_IWRITE</code>	<code>0x0002</code>
Owner execute	<code>S_IEXEC</code>	<code>0x0004</code>
Search permission	<code>S_ISEARCH</code>	<code>0x0004</code>

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

Description	Macro	EditMod
Group read	<code>S_IGREAD</code>	<code>0x0010</code>
Group write	<code>S_IGWRITE</code>	<code>0x0020</code>
Group execute	<code>S_IGEXEC</code>	<code>0x0040</code>
Group search	<code>S_IGSEARCH</code>	<code>0x0040</code>
Public read	<code>S_IOREAD</code>	<code>0x0100</code>
Public write	<code>S_IOWRITE</code>	<code>0x0200</code>
Public execute	<code>S_IOEXEC</code>	<code>0x0400</code>
Public search	<code>S_IOSEARCH</code>	<code>0x0400</code>



**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`).

### **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).

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

**Table 5-11 RBF Path Option Fields (continued)**

Field	Description File Macro
pd_wpc	PRECOMP
pd_rwr	REDWRITE
pd_park	PARK
pd_lsnooffs	LSNOFFS
pd_xfersize	XFERSIZE

### **EditMod Labels**

3-RBF path options  
1-number of surfaces

### **Description**

Indicates the number of surfaces (heads or sides) for a disk unit.

### **Port Generic Default Value**

2

### **Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

### **Available Values**

-2147483648 to 2147483647

EditMod Labels

3-RBF path options  
2-verify disk writes (0=verify)

Description

Indicates whether a write is verified by a re-read and compare. Write verify operations are generally performed on floppy disks but not hard disks because of the lower soft error rate of hard disks.

Port Generic Default Value

0 (zero)

Port Specific Override Value

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values

Device verify values are defined in the header file, rbf.h, and in Table 5-12.

Table 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



**Table 5-13** `pd_format` Available Values (continued)

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

### **EditMod Labels**

3-RBF path options  
4-number of cylinders

### **Description**

Indicates the number of cylinders per disk.

### **Port Generic Default Value**

80

### **Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

### **Available Values**

-2147483648 to 2147483647

**EditMod Labels**

3-RBF path options  
5-default blocks/track

**Description**

Indicates the number of blocks per track on the disk for all tracks except track 0. (See [pd\\_t0b](#) for track 0 information.)

**Port Generic Default Value**

2048

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

**Available Values**

-2147483648 to 2147483647

### **EditMod Labels**

3-RBF path options  
6-default blocks/track for trk0

### **Description**

Indicates the number of blocks per track 0 on the disk. Depending on the device, this may be a different number for track 0 than the other tracks on the disk.

### **Port Generic Default Value**

10

### **Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

### **Available Values**

-2147483648 to 2147483647

**EditMod Labels**

3-RBF path options  
7-segment allocation size

**Description**

This value specifies the default minimum number of sectors to be allocated when a file is expanded.

**Port Generic Default Value**

1

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

**Available Values**

-2147483648 to 2147483647

## 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

**EditMod Labels**

3-RBF path options  
9-track base offset

**Description**

This is the offset to the first accessible track number. Because Track 0 is often a different density, Track 0 is sometimes not used as the base track.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

**Available Values**

-2147483648 to 2147483647

**EditMod Labels**

3-RBF path options  
10-block base offset

**Description**

This is the offset to the first accessible sector number. Because Sector 0 is not always the base sector.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

**Available Values**

-2147483648 to 2147483647



**EditMod Labels**

3-RBF path options  
11-# tries

**Description**

This is the number of times a device tries to access a disk before returning an error.

**Port Generic Default Value**

7

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

**Available Values**

-2147483648 to 2147483647

### EditMod Labels

3-RBF path options  
12-size of block in bytes

### Description

This is the logical block size in bytes.

### Port Generic Default Value

256 (256 characters)

### Port Specific Override Value

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

### Available Values

-2147483648 to 2147483647

EditMod Labels

3-RBF path options  
13-control word

Description

This is the device control word.

Port Generic Default Value

Macro

CTRL\_AUTOSIZE

EditMod

0x2

Port Specific Override Value

Refer to RBF/<DEVICE>/DESC/config.des (Figure 5-3).

Available Values

Control word values are defined in the header file, rbf.h, and in Table 5-14.

Table 5-14 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

**Table 5-14** `pd_cntl` Available Values (continued)

Description	Macro	EditMod
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

**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

### **EditMod Labels**

3-RBF path options  
15-first reduced write current cylinder

### **Description**

This number indicates at which cylinder to begin reduced write current. Only older disk drives require this information, such as MFM or RLL drives.

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

### **Available Values**

-2147483648 to 2147483647

**EditMod Labels**

3-RBF path options  
16-park cylinder for hard disks

**Description**

This is the cylinder where the hard disk heads should be parked when the drive is shut down.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

**Available Values**

-2147483648 to 2147483647

### **EditMod Labels**

3-RBF path options  
17- lsn offset for partition

### **Description**

This is the offset to be used when accessing a partitioned drive.

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

### **Available Values**

-2147483648 to 2147483647



**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

**EditMod Labels**

4-RBF logical unit static storage  
1-interrupt vector

**Description**

This is the vector number of the device interrupt.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

**Available Values**

0 to 255

### **EditMod Labels**

4-RBF logical unit static storage  
2-interrupt level

### **Description**

This is the hardware priority of the device interrupt.

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

### **Available Values**

-128 to 127

**EditMod Labels**

4-RBF logical unit static storage  
3-interrupt priority

**Description**

This is the software (polling) priority of the device interrupt.

**Port Generic Default Value**

5

**Port Specific Override Value**

Refer to RBF/<DEVICE>/DESC/config.des (**Figure 5-3**).

**Available Values**

-128 to 127

# RBF Logical Unit Options

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

**Table 5-16 RBF Logical Unit Options Fields**

Field	Description File Macro
<code>lu_stp</code>	STEP
<code>lu_tfm</code>	DMAMODE
<code>lu_lun</code>	SCSILUN
<code>lu_ctrlrid</code>	CTRLRID
<code>lu_totcyls</code>	TOTCYLS

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

**Table 5-17** `lu_stp` Available Values (continued)

Description	Macro	EditMod
12 millisecond step rate	STEP_12MS	0x02
6 millisecond step rate	STEP_6MS	0x03



**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

### **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

---

# Chapter 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**



MICROWARE SOFTWARE

# 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

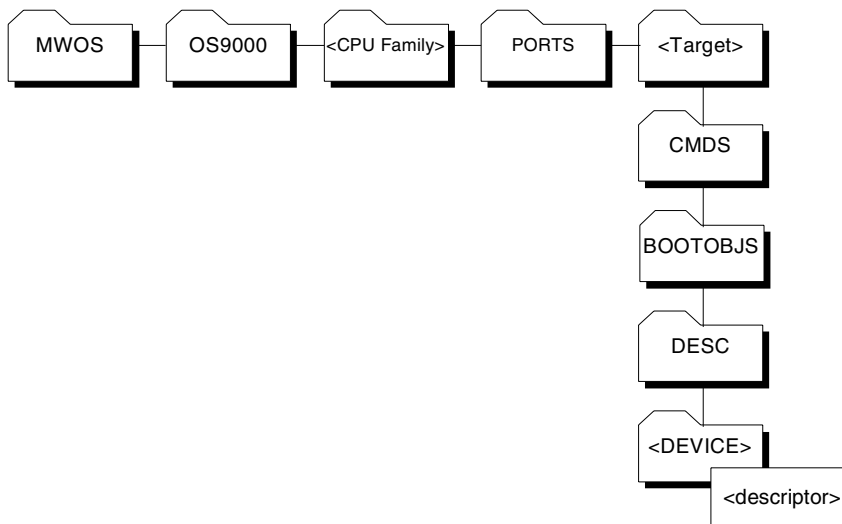
Use the `Editmod` utility and the following procedures to directly modify fields in the existing PCF device descriptor module. The module can stand-alone or it may be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The 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.



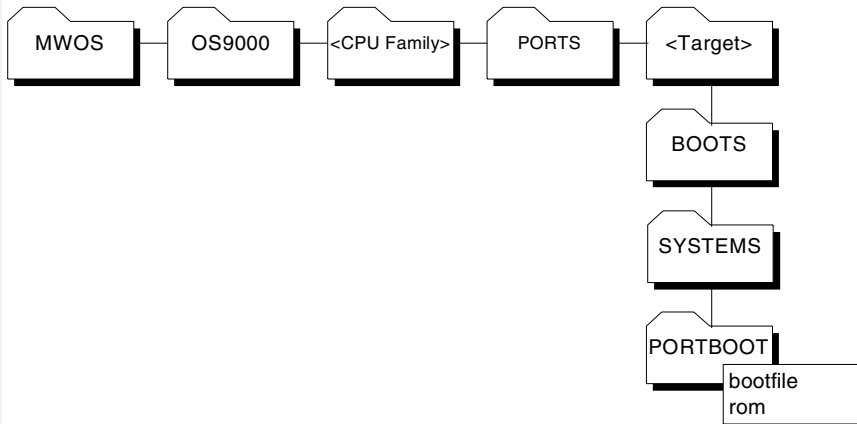
## For More Information

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**


---

## For More Information

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:

- 
- Step 1. Change to the `CMDS/BOOTOBJS/DESC/<DEVICE>` directory (see [Figure 6-1](#)).
  - Step 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:

- 
- Step 1. Change to the `BOOTS/SYSTEMS/PORTBOOT` directory (see [Figure 6-2](#)).
  - Step 2. Use EditMod to edit the module:  

```
$EditMod -e <descriptor> -f=<boot image name>
```
  - Step 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.
  - Step 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.
  - Step 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.
  - Step 6. Select the `w` command (write) to save the changes.
  - Step 7. Select the `q` command (quit) to exit `EditMod`.
- 



## Note

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

```

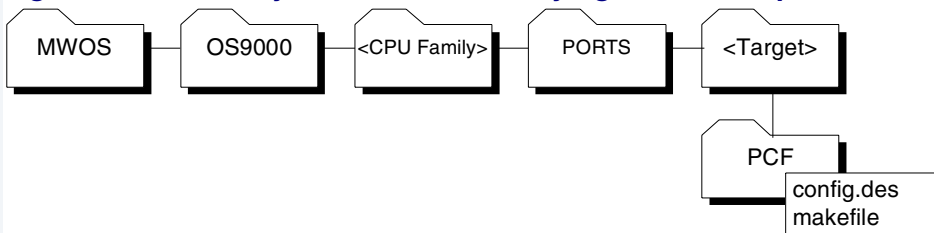
Which? [?/1-19/p/t/a/w/q] w

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

- 
- Step 1. Change to the `PCF/<DEVICE>` directory (see [Figure 6-3](#)).
  - Step 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.
  - Step 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.
  - Step 4. Read the comments in `config.des` to determine where to place the define for this macro.
  - Step 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>
```
  - Step 6. Save the changes and rebuild the PCF device descriptors, entering the following command in the `PCF/<DEVICE>/DESC` directory:  
  

```
os9make
```
  - Step 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>	<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 PCF/<DEVICE>/DESC/config.des ([Figure 6-3](#)).

**Available Values**

0 to 65535

### **EditMod Labels**

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

### **Description**

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

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

### **Available Values**

0 to 65535



**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).

## 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

**Table 6-2** `m_access` Available Values (continued)

Description	Macro	EditMod
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
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_SYSTEM_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 `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
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

**Table 6-3** `m_tylan` Available Module Type Values (continued)

Description	Macro	EditMod
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**.

**Note**

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\_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)
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>	<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>dd_port</code>	<code>MFGR_NAME</code>
<code>drvr_name</code>	<code>DRVR_NAME</code>
<code>dd_class</code>	<code>DD_CLASS</code>

**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

### **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

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

**Table 6-7** `dd_type` Available Values (continued)

Description	Macro	EditMod
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
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

## 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>
Append to file	<code>S_IAPPEND</code>	<code>0x1000</code>
Non-sharable	<code>S_ISHARE</code>	<code>0x4000</code>

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

Description	Macro	EditMod
Mask for permission bits	<code>S_IPRM</code>	<code>0xffff</code>
Owner read	<code>S_IREAD</code>	<code>0x0001</code>
Owner write	<code>S_IWRITE</code>	<code>0x0002</code>
Owner execute	<code>S_IEXEC</code>	<code>0x0004</code>
Search permission	<code>S_ISEARCH</code>	<code>0x0004</code>

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

Description	Macro	EditMod
Group read	<code>S_IGREAD</code>	<code>0x0010</code>
Group write	<code>S_IGWRITE</code>	<code>0x0020</code>
Group execute	<code>S_IGEXEC</code>	<code>0x0040</code>
Group search	<code>S_IGSEARCH</code>	<code>0x0040</code>
Public read	<code>S_IOREAD</code>	<code>0x0100</code>
Public write	<code>S_IOWRITE</code>	<code>0x0200</code>
Public execute	<code>S_IOEXEC</code>	<code>0x0400</code>
Public search	<code>S_IOSEARCH</code>	<code>0x0400</code>

**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**  
DRVR\_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`).

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

**Table 6-11 PCF Path Option Fields (continued)**

Field	Description File Macro
pd_wpc	PRECOMP
pd_rwr	REDWRITE
pd_park	PARK
pd_lsnooffs	LSNOFFS
pd_xfersize	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



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

**Table 6-13** `pd_format` Available Values (continued)

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

### **EditMod Labels**

3-PCF path options  
4-number of cylinders

### **Description**

Indicates the number of cylinders per disk.

### **Port Generic Default Value**

80

### **Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

### **Available Values**

-2147483648 to 2147483647

**EditMod Labels**

3-PCF path options  
5-default blocks/track

**Description**

Indicates the number of blocks per track on the disk for all tracks except track 0. (See [pd\\_t0b](#) for track 0 information.)

**Port Generic Default Value**

16

**Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

**Available Values**

-2147483648 to 2147483647

### **EditMod Labels**

3-PCF path options  
6-default blocks/track for trk0

### **Description**

Indicates the number of blocks per track 0 on the disk. Depending on the device, this can be a different number for track 0 than the other tracks on the disk.

### **Port Generic Default Value**

10

### **Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

### **Available Values**

-2147483648 to 2147483647

**EditMod Labels**

3-PCF path options  
7-segment allocation size

**Description**

This value specifies the default minimum number of sectors to be allocated when a file is expanded.

**Port Generic Default Value**

1

**Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

**Available Values**

-2147483648 to 2147483647

### EditMod Labels

3-PCF path options  
8-block interleave offset

### Description

This value determines the sector interleave factor. Sectors are arranged on a disk in a certain sequential order (1, 2, 3, ... or 1, 3, 5, ...). The interleave factor determines the arrangement. For example, if the interleave factor is 2, the sectors would be arranged by twos, (1,3,5,...) starting at the base sector. (See [pd\\_boffs](#) for base sector information.)

### Port Generic Default Value

3

### Port Specific Override Value

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

### Available Values

-2147483648 to 2147483647



**EditMod Labels**

3-PCF path options  
9-track base offset

**Description**

This is the offset to the first accessible track number. Because Track 0 is often a different density, Track 0 is sometimes not used as the base track.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

**Available Values**

-2147483648 to 2147483647

### **EditMod Labels**

3-PCF path options  
10-block base offset

### **Description**

This is the offset to the first accessible sector number. Because Sector 0 is not always the base sector.

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

### **Available Values**

-2147483648 to 2147483647

**EditMod Labels**

3-PCF path options

11-# tries

**Description**

This is the number of times a device tries to access a disk before returning an error.

**Port Generic Default Value**

7

**Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

**Available Values**

-2147483648 to 2147483647

### **EditMod Labels**

3-PCF path options  
12-size of block in bytes

### **Description**

This is the logical block size in bytes.

### **Port Generic Default Value**

256 (256 characters)

### **Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

### **Available Values**

-2147483648 to 2147483647

EditMod Labels

3-PCF path options  
13-control word

Description

This is the device control word.

Port Generic Default Value

Macro

CTRL\_MULTI

EditMod

0x1

Port Specific Override Value

Refer to PCF/<DEVICE>/DESC/config.des (Figure 6-3).

Available Values

Control word values are defined in the header file, PCF.h, and in Table 6-14.

Table 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

**Table 6-14** `pd_cntl` Available Values (continued)

Description	Macro	EditMod
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

**EditMod Labels**

3-PCF path options  
14-first write precomp cylinder

**Description**

This number indicates at which cylinder to begin write precompensation. Only older disk drives require this information, such as MFM or RLL drives.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

**Available Values**

-2147483648 to 2147483647

### **EditMod Labels**

3-PCF path options  
15-first reduced write current cylinder

### **Description**

This number indicates at which cylinder to begin reduced write current. Only older disk drives require this information, such as MFM or RLL drives.

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

### **Available Values**

-2147483648 to 2147483647



**EditMod Labels**

3-PCF path options  
16-park cylinder for hard disks

**Description**

This is the cylinder where the hard disk heads should be parked when the drive is shut down.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

**Available Values**

-2147483648 to 2147483647

### **EditMod Labels**

3-PCF path options  
17- lsn offset for partition

### **Description**

This is the offset to be used when accessing a partitioned drive.

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

### **Available Values**

-2147483648 to 2147483647

**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

### **EditMod Labels**

4-PCF logical unit static storage  
2-interrupt level

### **Description**

This is the hardware priority of the device interrupt.

### **Port Generic Default Value**

3

### **Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

### **Available Values**

-128 to 127

**EditMod Labels**

4-PCF logical unit static storage  
3-interrupt priority

**Description**

This is the software (polling) priority of the device interrupt.

**Port Generic Default Value**

10

**Port Specific Override Value**

Refer to PCF/<DEVICE>/DESC/config.des (**Figure 6-3**).

**Available Values**

-128 to 127

# PCF Logical Unit Options

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

**Table 6-16 PCF Logical Unit Options Fields**

Field	Description File Macro
<code>lu_stp</code>	STEP
<code>lu_tfm</code>	DMAMODE
<code>lu_lun</code>	SCSILUN
<code>lu_ctrlrid</code>	CTRLRID
<code>lu_totcyls</code>	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

**Table 6-17** `lu_stp` Available Values (continued)

Description	Macro	EditMod
12 millisecond step rate	STEP_12MS	0x02
6 millisecond step rate	STEP_6MS	0x03

**EditMod Labels**

4-PCF logical unit static storage  
4-PCF logical unit options  
2-dma transfer mode

**Description**

This hardware specific byte can be set for use of DMA mode, if it is available. DMA requires only a single interrupt for each block of characters transferred in an I/O operation. It is much faster than methods that interrupt for each character transferred.

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

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

**Available Values**

-128 to 127

### **EditMod Labels**

4-PCF logical unit static storage  
4-PCF logical unit options  
3-drive logical unit number

### **Description**

This number is used in the command block to identify the drive to the controller. The driver uses this number when specifying the device.

### **Port Generic Default Value**

0 (zero)

### **Port Specific Override Value**

Refer to 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

### **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

---

# Chapter 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**



MICROWARE SOFTWARE

# 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

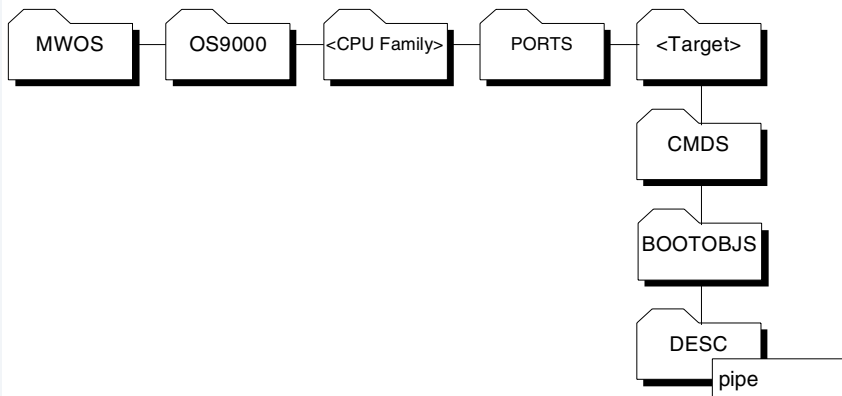
Use the `Editmod` utility and the following procedures to directly modify fields in the existing pipe device descriptor module. The module can stand-alone or it can be part of a merged-module group. A boot image, for example, contains multiple modules. Both situations are covered in this section. The 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.



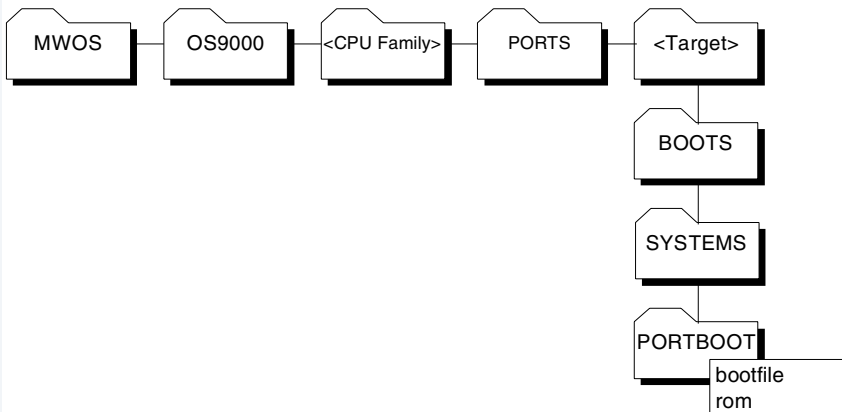
## For More Information

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**





---

## For More Information

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:

---

Step 1. Change to the `CMDS/BOOTBJS/DESC/<DEVICE>` directory (see **Figure 7-1**).

Step 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:

---

Step 1. Change to the `BOOTS/SYSTEMS/PORTBOOT` directory (see **Figure 7-2**).

Step 2. Use EditMod to edit the module:

```
$EditMod -e <descriptor> -f=<boot image name>
```

Step 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.

Step 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.

- Step 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.
- Step 6. Select the `w` command (write) to save the changes.
- Step 7. Select the `q` command (quit) to exit EditMod.



## Note

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
```

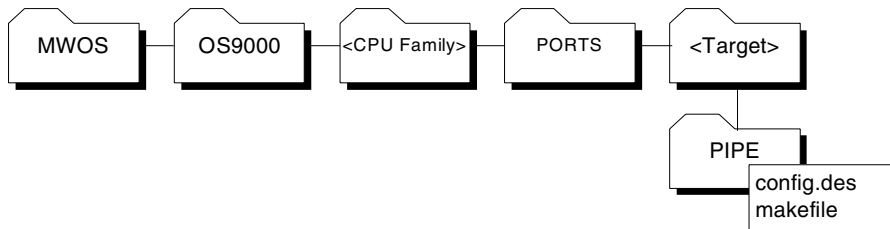
```
Which? [?/1-19/p/t/a/w/q] w
```

```
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

- 
- Step 1. Change to the `PIPE/<DEVICE>` directory (see [Figure 7-3](#)).
- Step 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.
- Step 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.
- Step 4. Read the comments in `config.des` to determine where to place the define for this macro.
- Step 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>
```
- Step 6. Save the changes and rebuild the pipe device descriptors, entering the following command in the `PIPE/<DEVICE>/DESC` directory:
- ```
os9make
```
- Step 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

### **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`).

### 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  |
| 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 |
| 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  |

**Table 7-3** `m_tylan` Available Module Type Values (continued)

| Description      | Macro     | EditMod   |
|------------------|-----------|-----------|
| 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\_attrv****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](#).

**Note**

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<br>(shifted left to first byte:<br>MA_REENT<<8) | 0x80<br>(shifted left to first byte:<br>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<br>(shifted left to first byte:<br>MA_GHOST<<8) | 0x40<br>(shifted left to first byte:<br>0x4000) |
| The module is a system-state module.                                                                                                    | MA_SUPER<br>(shifted left to first byte:<br>MA_SUPER<<8) | 0x20<br>(shifted left to first byte:<br>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> |
| <code>drvr_name</code>  | <code>DRVR_NAME</code> |
| <code>dd_class</code>   | <code>DD_CLASS</code>  |

**EditMod Labels**

2-device descriptor data definitions  
1-device port address

**Description**

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

**Macro Example**

```
#define PORTADDR    0xfffe4000
```

**Port Generic Default Value**

0 (zero)

**Port Specific Override Value**

Refer to PIPE/config.des (**Figure 7-3**).

**Available Values**

0 to 4294967295

### **EditMod Labels**

2-device descriptor data definitions  
2-logical unit number

### **Description**

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

### **Macro Example**

```
#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

### 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     |



**Table 7-7** `dd_type` Available Values (continued)

| Description                     | Macro    | EditMod   |
|---------------------------------|----------|-----------|
| 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       |
| 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 |

### 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> |
| Append to file                 | <code>S_IAPPEND</code> | <code>0x1000</code> |
| Non-sharable                   | <code>S_ISHARE</code>  | <code>0x4000</code> |

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

| Description              | Macro                 | EditMod             |
|--------------------------|-----------------------|---------------------|
| Mask for permission bits | <code>S_IPRM</code>   | <code>0xffff</code> |
| Owner read               | <code>S_IREAD</code>  | <code>0x0001</code> |
| Owner write              | <code>S_IWRITE</code> | <code>0x0002</code> |
| Owner execute            | <code>S_IEXEC</code>  | <code>0x0004</code> |

**Table 7-9** `dd_mode` Available Values for File Access Permissions  
(continued)

| Description       | Macro                   | EditMod             |
|-------------------|-------------------------|---------------------|
| Search permission | <code>S_ISEARCH</code>  | <code>0x0004</code> |
| Group read        | <code>S_IGREAD</code>   | <code>0x0010</code> |
| Group write       | <code>S_IGWRITE</code>  | <code>0x0020</code> |
| Group execute     | <code>S_IGEXEC</code>   | <code>0x0040</code> |
| Group search      | <code>S_IGSEARCH</code> | <code>0x0040</code> |
| Public read       | <code>S_IOREAD</code>   | <code>0x0100</code> |
| Public write      | <code>S_IOWRITE</code>  | <code>0x0200</code> |
| Public execute    | <code>S_IOEXEC</code>   | <code>0x0400</code> |
| Public search     | <code>S_IOSEARCH</code> | <code>0x0400</code> |

**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\_name**  
DRV\_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).

**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> | <code>BUFSZ</code>     |



**EditMod Labels**

3-pipeman logical unit static storage  
1-pipe FIFO buffer size

**Description**

Used to define the buffer size of the pipe.

**Port Generic Default Value**

256

**Port Specific Override Value**

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

**Available Values**

0 to 4294967295



# Index

## A

- access
  - changing
    - for init [159](#)
- access permissions
  - setting
    - for cnfgdata [21](#)
    - for init [113](#)
    - for PCF [522](#)
    - for pipe [588](#)
    - for RBF [456](#)
    - for SBF [412](#)
    - for SCF [182](#)
- acct\_name
  - changing
    - for init [136](#)
- attributes
  - of module
    - setting for cnfgdata [26](#)
    - setting for init [118](#)
    - setting for PCF [527](#)
    - setting for pipe [593](#)
    - setting for RBF [461](#)
    - setting for SBF [417](#)
    - setting for SCF [187](#)
- autoboot\_delay
  - changing
    - for cnfgdata [95](#)
- AUTOECHO
  - changing
    - for SCF [388](#)
- AUTOLF
  - changing
    - for SCF [390](#)

---

**B**

B\_NVRAM [160](#)  
B\_PARITY [160](#)  
B\_ROM [160](#)  
B\_SHARED [160](#)  
B\_USERRAM [160](#)  
back space character  
    changing [384](#)  
baud rate  
    changing for SCF device [226](#), [229](#)  
BAUDRATE  
    changing  
        for SCF [226](#)  
bell character  
    changing [380](#)  
BELLCH  
    changing  
        for SCF [380](#)  
bits per character,  
    changing the [232](#)  
blk\_beg  
    changing  
        for init [167](#)  
blk\_end  
    changing  
        for init [168](#)  
BLKOFFS  
    changing  
        for PCF [554](#)  
        for RBF [488](#)  
blksiz  
    changing  
        for init [161](#)  
BLKSIZE  
    changing [490](#), [556](#)  
        for SBF [433](#)  
BLKSTRK  
    changing  
        for PCF [549](#)  
        for RBF [483](#)

**BLKSTRK0**

changing

for PCF 550

for RBF 484

block offset

changing 488, 489, 490, 491, 554, 555, 556, 557

block size 433

blocks per track

number of on disk

changing 483, 549

blocks per track 0

number of on disk

changing 484, 550

boot data

boot\_abname

configuration 90

boot\_automenu

configuration 93

boot\_delay

configuration 95

boot\_newab

configuration 91

boot\_newname

configuration 92

boot\_params

configuration 94

boot\_abname

boot data

configuration 90

changing

for cnfgdata 90

boot\_automenu

boot data

configuration 93

changing

for cnfgdata 93

**BOOT\_CMDSIZE**

changing 88

boot\_cmdsize

changing

for cnfgdata 88

BOOT\_COUNT  
    changing 87  
boot\_count  
    changing  
        for cnfgdata 87  
boot\_delay  
    boot data  
        configuration 95  
boot\_newab  
    boot data  
        configuration 91  
    changing  
        for cnfgdata 91  
boot\_newname  
    boot data  
        configuration 92  
    changing  
        for cnfgdata 92  
boot\_params  
    boot data  
        configuration 94  
    changing  
        for cnfgdata 94  
brdcst\_address  
    changing  
        for cnfgdata 75  
    interface data  
        configuration 75  
BSB  
    changing  
        got SCF 384  
BSPCH  
    changing  
        for SCF 381  
buffer size  
    changing  
        for pipe 609  
BUFSZ  
    changing  
        for pipe 609  
bufsz

changing  
for pipe 609

---

## C

### C I-code (reserved)

#### module header

language code 25, 117  
language code for PCF 526  
language code for pipe 592  
language code for RBF 460  
language code for SBF 416  
language code for SCF 186

chd utility 130

chx utility 130

#### cinit

##### changing

m\_attrev (attributes/revision) field 118

#### cnfgdata

##### changing

autoboot\_delay field 95  
boot\_abname field 90  
boot\_automenu field 93  
boot\_cmdsize field 88  
boot\_count field 87  
boot\_newab field 91  
boot\_newname field 92  
boot\_params field 94  
brdcst\_address field 75  
communication device cons\_baudrate field 55  
communication device cons\_flow field 62  
communication device cons\_level field 51  
communication device cons\_parity field 53  
communication device cons\_priority field 49  
communication device cons\_stopbits field 60  
communication device cons\_timeout field 52  
communication device cons\_vector field 48  
communication device cons\_wordsize field 58  
cons\_name field 30  
console device cons\_baudrate field 38  
console device cons\_flow field 45

console device cons\_level field 34  
 console device cons\_parity field 36  
 console device cons\_priority field 32  
 console device cons\_stopbits field 43  
 console device cons\_timeout field 35  
 console device cons\_vector field 31  
 console device cons\_wordsize field 41  
 debug\_call\_at\_cold field 65  
 debugger\_name field 64  
 gw\_address field 76  
 hwtype field 78  
 if\_flags field 79  
 if\_level field 85  
 if\_name field 80  
 if\_priority field 83  
 if\_vector field 82  
 ip\_address field 73  
 lpm\_count field 70  
 m\_access field 21  
 m\_attrv (attributes/revision) field 26  
 m\_edit field 28  
 m\_group field 18  
 m\_tylan (type/language) field 23  
 m\_user field 19  
 mac\_Address field 77  
 max\_notifiers field 97  
 maxllpmconns field 69  
 maxllpmprotos field 67  
 maxrcvmbufs field 68  
 port\_address field 81  
 subnet\_mask field 74  
 changing mod\_name field 20  
 MH\_ACCESS  
     changing 21  
 MH\_EDITION  
     changing 28  
 MH\_GROUP  
     changing 18  
 MH\_NAME  
     changing 20  
 MH\_TYLAN



- changing 23
- MH\_USER
  - changing 19
- module header
  - group ID 18
  - user ID
    - module header 19
- COBOL I-code (reserved)
  - module header
    - language code 25, 117
    - language code for PCF 526
    - language code for pipe 592
    - language code for RBF 460
    - language code for SBF 416
    - language code for SCF 186
- COMM\_BAUDRATE
  - changing 55
- COMM\_FLOW
  - changing 62
- COMM\_PARITY
  - changing 53
- COMM\_PRIORITY
  - changing 49
- COMM\_STOPBITS
  - changing 60
- COMM\_TIMEOUT
  - changing 52
- COMM\_VECTOR
  - changing 48
- COMM\_WORDSIZE
  - changing 58
- COMPAT
  - changing
    - for init 152
- configuration data block
  - module header
    - type code for cnfgdata 24
    - type code for init 116
    - type code for PCF 525
    - type code for pipe 591
    - type code for RBF 459

- type code for SBF 415
  - type code for SCF 185
- configuration module 107
- CONS\_BAUDRATE
  - changing 38
- cons\_baudrate
  - changing
    - for cnfgdata communication device 55
    - for cnfgdata console device 38
- CONS\_FLOW
  - changing 45
- cons\_flow
  - changing
    - for cnfgdata communication device 62
    - for cnfgdata console device 45
- CONS\_LEVEL
  - changing 34
- cons\_level
  - changing
    - for cnfgdata communication device 51
    - for cnfgdata console device 34
- CONS\_NAME
  - changing 30
  - for init 131
- CONS\_PARITY
  - changing 36
- cons\_parity
  - changing
    - for cnfgdata communication device 53
    - for cnfgdata console device 36
- CONS\_PRIORITY
  - changing 32
- cons\_priority
  - changing
    - for cnfgdata communication device 49
    - for cnfgdata console device 32
- CONS\_STOPBITS
  - changing 43
- cons\_stopbits
  - changing
    - for cnfgdata communication device 60

- for cnfgdata console device 43
- CONS\_TIMEOUT
  - changing 35
- cons\_timeout
  - changing
    - for cnfgdata communication device 52
    - for cnfgdata console device 35
- CONS\_VECTOR
  - changing 31
- cons\_vector
  - changing
    - for cnfgdata communication device 48
    - for cnfgdata console device 31
- CONS\_WORDSIZE
  - changing 41
- cons\_wordsize
  - changing
    - for cnfgdata communication device 58
    - for cnfgdata console device 41
- console device
  - setting vector number 31, 48
- console device name
  - setting 30
- console\_name
  - changing
    - for cnfgdata 30
    - for init 131
- CONTROL
  - changing 491, 557
- controller ID number
  - setting 507, 573, 574
- CPUCOMPAT
  - changing
    - for init 146
- CTRLRID
  - changing 507, 573
  - for PCF 574
- cylinder
  - starting reduced write
    - changing 494, 560
- cylinders

number of disk  
changing 482, 548  
CYLNDRS  
changing 482, 548

D

data module  
module header  
type code for cnfgdata 24  
type code for init 116  
type code for PCF 525  
type code for pipe 591  
type code for RBF 459  
type code for SBF 415  
type code for SCF 185  
DC\_RND  
dd\_class  
available value 202, 431, 475, 541, 607  
DC\_SEQ  
dd\_class  
available value 202, 431, 475, 541, 607  
DD\_CLASS  
changing  
for PCF 541  
for pipe 607  
for RBF 475  
for SBF 431  
for SCF 202  
dd\_class  
available value  
DC\_RND 202, 431, 475, 541, 607  
DC\_SEQ 202, 431, 475, 541, 607  
changing  
for PCF 541  
for pipe 607  
for RBF 475  
for SBF 431  
for SCF 202  
dd\_lun  
changing

for PCF 532  
 for pipe 598  
 for RBF 466  
 for SBF 422  
 for SCF 192

## DD\_MODE

### changing

for PCF 536  
 for pipe 602  
 for RBF 470  
 for SBF 426  
 for SCF 197

## dd\_mode

### available value

S\_IAPPEND 198, 427, 471, 537, 603  
 S\_ICONTIG 198, 427, 471, 537, 603  
 S\_ICREAT 198, 427, 471, 537, 603  
 S\_IEXCL 198, 427, 471, 537, 603  
 S\_IEXE 198, 427, 471, 537, 603  
 S\_IGEXEC 199, 428, 472, 538, 604  
 S\_IGREAD 199, 428, 472, 538, 604  
 S\_IGSEARCH 199, 428, 472, 538, 604  
 S\_IGWRITE 199, 428, 472, 538, 604  
 S\_IOEXEC 199, 428, 472, 538, 604  
 S\_IOREAD 199, 428, 472, 538, 604  
 S\_IOSEARCH 199, 428, 472, 538, 604  
 S\_IOWRITE 199, 428, 472, 538, 604  
 S\_IPRM 198, 427, 471, 537, 603  
 S\_IREAD 198, 427, 471, 537, 603  
 S\_ISEARCH 198, 427, 471, 537, 604  
 S\_ISHARE 198, 427, 471, 537, 603  
 S\_ITRUNC 198, 427, 471, 537, 603  
 S\_IWRITE 198, 427, 471, 537, 603

### changing

for PCF 536  
 for pipe 602  
 for RBF 470  
 for SBF 426  
 for SCF 197

## dd\_pd\_size

### changing

- for PCF [533](#)
  - for pipe [599](#)
  - for RBF [467](#)
  - for SBF [423](#)
  - for SCF [193](#)
- dd\_port
  - changing
    - for PCF [531](#)
    - for pipe [597](#)
    - for RBF [465](#)
    - for SBF [421](#)
    - for SCF [191](#)
- DD\_TYPE
  - changing
    - for PCF [534](#)
    - for pipe [600](#)
    - for RBF [468](#)
    - for SBF [424](#)
    - for SCF [194](#)
- dd\_type
  - available values
    - DT\_RBF [194](#), [424](#), [468](#), [534](#), [600](#)
    - DT\_SCF [194](#), [424](#), [468](#), [534](#), [600](#)
  - changing
    - for PCF [534](#)
    - for pipe [600](#)
    - for RBF [468](#)
    - for SBF [424](#)
    - for SCF [194](#)
- debug\_call\_at\_cold
  - changing
    - for cnfgdata [65](#)
- debug\_name
  - changing
    - for cnfgdata [64](#)
- DEBUGGER\_COLD\_FLAG
  - changing [65](#)
- DEBUGGER\_NAME
  - changing [64](#)
- desc
  - changing

- for init [164](#)
- DEV\_ERASE\_FLG [434](#)
- DEV\_OFFLINE\_FLG [434](#)
- DEV\_REWIND\_FLG [434](#)
- DEV\_SKIPBACK\_FLG [434](#)
- device
  - identify unique
    - for controller [192](#), [422](#), [466](#), [532](#), [598](#)
- Device Controller SCSI ID
  - changing [436](#)
- device descriptor
  - module header
    - type code for cnfgdata [24](#)
    - type code for init [116](#)
    - type code for PCF [525](#)
    - type code for pipe [591](#)
    - type code for RBF [459](#)
    - type code for SBF [415](#)
    - type code for SCF [185](#)
- device driver
  - module header
    - type code for cnfgdata [24](#)
    - type code for init [116](#)
    - type code for PCF [525](#)
    - type code for pipe [591](#)
    - type code for RBF [459](#)
    - type code for SBF [415](#)
    - type code for SCF [185](#)
  - name string
    - selecting for PCF [540](#)
    - selecting for pipe [606](#)
    - selecting for RBF [474](#)
    - selecting for SBF [430](#)
    - selecting for SCF [201](#)
  - port address
    - for hardware controller [191](#), [421](#), [465](#), [531](#), [597](#)
- device interrupt
  - changing [440](#), [500](#), [566](#)
- device mode
  - I/O class of [197](#), [426](#), [470](#), [536](#), [602](#)
- device state

- changing 442
- device type
  - I/O class of 194, 424, 468, 534, 600
- DFLG\_DRIVEBUSY 442
- DFLG\_EOFFLAG 442
- DFLG\_READFLAG 442
- DFLG\_WRITEFLAG 442
- disk cylinders
  - changing 482, 548
- disk format
  - changing 480, 546
- disk sides
  - changing 478, 544
- disk type
  - changing 480, 546
- disk write verification
  - changing 479, 545
- DMA mode
  - changing 435
- DMA transfer mode
  - setting 505, 571
- dma\_addr
  - changing
    - for init 165
- DMAMODE
  - changing
    - for PCF 571
    - for RBF 505
    - for SBF 435
- drive unit number
  - setting 506, 572
- DRIVE\_FLAG
  - changing 442
- drive\_name
  - changing
    - for init 130
- Driver Compatibility Flags
  - changing 434
- DRVR\_NAME
  - changing
    - for PCF 540



for pipe 606  
for RBF 474  
for SBF 430  
for SCF 201  
drvr\_name  
changing  
for PCF 540  
for pipe 606  
for RBF 474  
for SBF 430  
for SCF 201  
DSPTBLSZ  
changing  
for init 145  
DT\_CDFM 195, 425, 469, 535, 601  
DT\_GFM 195, 425, 469, 535, 601  
DT\_INET 195, 425, 469, 535, 601  
DT\_ISDN 195, 425, 469, 535, 601  
DT\_MPFM 195, 425, 469, 535, 601  
DT\_NFM 195, 425, 469, 535, 601  
DT\_NRF 195, 425, 469, 535, 601  
DT\_PCF 195, 425, 469, 535, 601  
DT\_PIPE 194, 424, 468, 534, 600  
DT\_PTTY 195, 425, 469, 535, 601  
DT\_RBF  
dd\_type  
available values 194, 424, 468, 534, 600  
DT\_RTNFM 196, 425, 469, 535, 601  
DT\_SBF 195, 425, 469, 535, 601  
DT\_SCF  
dd\_type  
available values 194, 424, 468, 534, 600  
DT\_SOCKET 195, 425, 469, 535, 601  
DT\_SPF 196, 425, 469, 535, 601  
DT\_UCM 195, 425, 469, 535, 601

E

echo character  
changing 388  
edition number

- of module
    - setting for cnfgdata 28
    - setting for init 120
    - setting for PCF 529
    - setting for pipe 595
    - setting for RBF 463
    - setting for SBF 419
    - setting for SCF 189
- end of file character
  - changing 378
- end of line character
  - changing 396, 397, 398
- end of record character
  - changing 375
- EOF character
  - changing 378
- EOFCH
  - changing
    - for SCF 378
- EOLNULLS
  - changing
    - for SCF 396
- EOR character
  - changing 375
- EORCH
  - changing
    - for SCF 375
- EVENTS
  - changing
    - for init 139
- executable
  - module header
    - type code for cnfgdata 23
    - type code for init 116
    - type code for PCF 525
    - type code for pipe 591
    - type code for RBF 459
    - type code for SBF 415
    - type code for SCF 185
- extens\_list
  - changing

for init 132  
EXTENSIONS  
changing  
for init 132

---

F

file  
sysboot 107  
file manager  
module header  
type code for cnfgdata 24  
type code for init 116  
type code for PCF 525  
type code for pipe 591  
type code for RBF 459  
type code for SBF 415  
type code for SCF 185  
name string  
selecting for PCF 539  
selecting for pipe 605  
selecting for RBF 473  
selecting for SBF 429  
selecting for SCF 200  
FLAGS  
changing 434  
FMGR\_NAME  
changing  
for PCF 539  
for pipe 605  
for RBF 473  
for SBF 429  
for SCF 200  
fmgr\_name  
changing  
for PCF 539  
for pipe 605  
for RBF 473  
for SBF 429  
for PCF  
changing

- PRIORITY 567
- for RBF
  - changing
    - PRIORITY 501
- for SBF
  - changing
    - DRIVE\_FLAG 442
    - PRIORITY 441
- FORMAT
  - changing
    - for PCF 546
    - for RBF 480
- format
  - of disk
    - changing 480, 546
- Fortran
  - module header
    - language code 25, 117
    - language code for PCF 526
    - language code for pipe 592
    - language code for RBF 460
    - language code for SBF 416
    - language code for SCF 186
- FUNC0x01
  - changing
    - for SCF 252
- FUNC0x02
  - changing
    - for SCF 256
- FUNC0x03
  - changing
    - for SCF 260
- FUNC0x04
  - changing
    - for SCF 264
- FUNC0x05
  - changing
    - for SCF 268
- FUNC0x06
  - changing
    - for SCF 272

|          |                     |
|----------|---------------------|
| FUNC0x07 |                     |
| changing |                     |
| for SCF  | <a href="#">276</a> |
| FUNC0x08 |                     |
| changing |                     |
| for SCF  | <a href="#">280</a> |
| FUNC0x09 |                     |
| changing |                     |
| for SCF  | <a href="#">284</a> |
| FUNC0x0a |                     |
| changing |                     |
| for SCF  | <a href="#">288</a> |
| FUNC0x0b |                     |
| changing |                     |
| for SCF  | <a href="#">292</a> |
| FUNC0x0c |                     |
| changing |                     |
| for SCF  | <a href="#">296</a> |
| FUNC0x0d |                     |
| changing |                     |
| for SCF  | <a href="#">300</a> |
| FUNC0x0e |                     |
| changing |                     |
| for SCF  | <a href="#">304</a> |
| FUNC0x0f |                     |
| changing |                     |
| for SCF  | <a href="#">308</a> |
| FUNC0x10 |                     |
| changing |                     |
| for SCF  | <a href="#">312</a> |
| FUNC0x11 |                     |
| changing |                     |
| for SCF  | <a href="#">316</a> |
| FUNC0x12 |                     |
| changing |                     |
| for SCF  | <a href="#">320</a> |
| FUNC0x13 |                     |
| changing |                     |
| for SCF  | <a href="#">324</a> |
| FUNC0x14 |                     |
| changing |                     |

|          |          |
|----------|----------|
| for SCF  | 328      |
| FUNC0x15 |          |
| changing |          |
| for SCF  | 332      |
| FUNC0x16 |          |
| changing |          |
| for SCF  | 336      |
| FUNC0x17 |          |
| changing |          |
| for SCF  | 340      |
| FUNC0x18 |          |
| changing |          |
| for SCF  | 344, 348 |
| FUNC0x1a |          |
| changing |          |
| for SCF  | 352      |
| FUNC0x1b |          |
| changing |          |
| for SCF  | 356      |
| FUNC0x1c |          |
| changing |          |
| for SCF  | 360      |
| FUNC0x1d |          |
| changing |          |
| for SCF  | 364      |
| FUNC0x1e |          |
| changing |          |
| for SCF  | 368      |
| FUNC0x1f |          |
| changing |          |
| for SCF  | 372      |
| FUNC0x7f |          |
| changing |          |
| for SCF  | 247      |

---

G

|                            |     |
|----------------------------|-----|
| ghost                      |     |
| module                     |     |
| setting value for cnfgdata | 27  |
| setting value for init     | 119 |

- setting value for PCF 528
  - setting value for pipe 594
  - setting value for RBF 462
  - setting value for SBF 418
  - setting value for SCF 188
- Greenwich Mean Time (GMT) 147
- group ID
  - module header
    - cnfgdata 18
    - init 110
    - PCF 519
    - pipe 585
    - RBF 453
    - SBF 409
    - SCF 179
- gw\_address
  - changing
    - for cnfgdata 76
  - interface data
    - configuration 76

---

H

- hardware controller
  - absolute physical address
    - for dd\_port 191, 421, 465, 531, 597
- hardware\_vector
  - changing
    - for SCF 207
- head step rate
  - changing 503, 569
- header files
  - io.h
    - for available device types 194, 424, 468, 534, 600
  - modes.h
    - setting dd\_mode 198, 427, 471, 537, 603
- hilim
  - changing
    - for init 163
- hwtype
  - changing

for cnfgdata 78  
interface data  
configuration 78

I

I/O class of  
device mode  
changing 197, 426, 470, 536, 602  
device type  
changing 194, 424, 468, 534, 600  
I\_CREATE  
setting access mode 197, 426, 470, 536, 602  
I\_OPEN  
setting access mode 197, 426, 470, 536, 602  
if\_flags  
changing  
for cnfgdata 79  
interface data  
configuration 79  
if\_level  
changing  
for cnfgdata 85  
interface data  
configuration 85  
if\_name  
changing  
for cnfgdata 80  
if\_name\_ether  
interface data  
configuration 80  
if\_name\_slip  
interface data  
configuration 80  
if\_priority  
changing  
for cnfgdata 83  
interface data  
configuration 83  
if\_vector  
changing



- for cnfgdata 82
- interface data
  - configuration 82
- init
  - changing
    - access field 159
    - acct\_name field 136
    - blk\_beg field 167
    - blk\_end field 168
    - blksiz field 161
    - COMPAT 152
    - CONS\_NAME 131
    - console\_name field 131
    - CPUCOMPAT 146
    - desc field 164
    - dma\_addr field 165
    - drive\_name field 130
    - DSPTBLSZ 145
    - EVENTS 139
    - extens\_list field 132
    - EXTENSIONS 132
    - hilim field 163
    - install\_name field 126
    - ioman\_name field 135
    - lolim field 162
    - m\_access field 113
    - m\_compat field 152
    - m\_cpucompat field 146
    - m\_cputyp field 125
    - m\_dsptbl field 145
    - m\_edit field 120
    - m\_edition field 151
    - m\_events field 139
    - m\_group field 110
    - m\_level field 148
    - m\_major field 149
    - m\_maxage field 144
    - m\_maxsigs field 154
    - m\_minor field 150
    - m\_minpty field 143
    - m\_paths field 138

m\_procs field 137  
 m\_site field 124  
 m\_slice field 141  
 m\_syspri field 142  
 m\_tmzone field 147  
 m\_tylan (type/language) field 115  
 m\_user field 111  
 MAXPTY 144  
 MAXSIGs 154  
 MINPTY 143  
 MPUCHIP 125  
 OS\_EDITION 151  
 OS\_LEVEL 148  
 OS\_REVISION 150  
 OS\_VERSION 149  
 OS9K\_REVSTR 127  
 os9rev\_name field 127  
 PATHS 138  
 preio\_name field 155  
 PREIOS 155  
 prior field 158  
 PROCS 137  
 RTC\_NAME 134  
 rtc\_name field 134  
 SITE 124  
 SLICE 141  
 sparam\_string field 129  
 SYS\_DEVICE 130  
 SYS\_PARAMS 129  
 SYS\_PRIOR 142  
 SYS\_START 128  
 SYS\_TMZONE 147  
 sysgo\_name field 128  
 TICK\_NAME 133  
 TICK\_SEC 140  
 ticker\_name field 133  
 ticsec field 140  
 type field 157  
 USRACCT\_NAME 136  
 changing IOMAN\_NAME 135  
 changing mod\_name field 112

INSTALNAME  
     changing 126  
 MH\_ACCESS  
     changing 113  
 MH\_EDITION  
     changing 120  
 MH\_GROUP  
     changing 110  
 MH\_NAME  
     changing 112  
 MH\_TYLAN  
     changing 115  
 MH\_USER  
     changing 111  
 module header  
     group ID 110  
     user ID  
         module header 111  
 Init module 107  
 init.h 108  
 INPUT\_TYPE  
     changing  
         for SCF 211  
 insert mode character  
     changing 394  
 INSERTMODE  
     changing  
         for SCF 394  
 INSIZE  
     changing  
         for SCF 216  
 install\_name  
     changing  
         for init 126  
 INSTALNAME  
     changing  
         for init 126  
 interface data  
     brdcst\_address  
         configuration 75  
     gw\_address

- configuration 76
- hwtype
  - configuration 78
- if\_flags
  - configuration 79
- if\_level
  - configuration 85
- if\_name\_ether
  - configuration 80
- if\_name\_slip
  - configuration 80
- if\_priority
  - configuration 83
- if\_vector
  - configuration 82
- ip\_address
  - configuration 73
- mac\_address
  - configuration 77
- port\_address
  - configuration 81
- subnet\_mask
  - configuration 74
- interleave factor
  - changing 486, 552
- intermediate code
  - module header
    - language code for cnfgdata 25
    - language code for init 117
    - language code for PCF 526
    - language code for pipe 592
    - language code for RBF 460
    - language code for SBF 416
    - language code for SCF 186
- interrupt control key
  - changing for keyboard 219
- interrupt levels
  - supported number
    - changing 208
- interrupt vector
  - setting

- for console device 31, 48
- INTRLV
  - changing
    - for PCF 552
    - for RBF 486
- io.h
  - available device types 194, 424, 468, 534, 600
- IOMAN\_NAME
  - changing
    - for init 135
- ioman\_name
  - changing
    - for init 135
- ip\_address
  - changing
    - for cnfgdata 73
  - interface data
    - configuration 73
- IRQ\_MASK
  - changing
    - for SCF 214
- IRQLEVEL
  - changing 440, 500, 566
  - for SCF 208

---

**K**

- keyboard interrupt
  - changing control key 219
- keyboard pause
  - changing control key 223
- keyboard quit
  - changing control key 222
- keyboard X-OFF
  - changing control key 225
- keyboard X-ON
  - changing control key 224
- KYBDINTR
  - changing
    - for SCF 219
- KYBDPAUSE

- changing
  - for SCF [223](#)
- KYBDQUIT
  - changing
    - for SCF [222](#)

---

**L**

- language
  - required for running
    - setting for cnfgdata [23](#)
    - setting for init [115](#)
    - setting for PCF [524](#)
    - setting for pipe [590](#)
    - setting for RBF [458](#)
    - setting for SBF [414](#)
    - setting for SCF [184](#)
- language code
  - available values
    - MT\_CBLCODE [25](#), [117](#)
    - MT\_CCODE [25](#), [117](#)
    - MT\_FRTNCODE [25](#), [117](#)
    - MT\_MASK [25](#), [117](#)
  - available values for cnfgdata
    - MT\_ANY [25](#)
    - MT\_ICODE [25](#)
    - MT\_OBJECT [25](#)
    - MT\_PCODE [25](#)
  - available values for init
    - MT\_ANY [117](#)
    - MT\_ICODE [117](#)
    - MT\_OBJECT [117](#)
    - MT\_PCODE [117](#)
  - available values for PCF
    - MT\_ANY [526](#)
    - MT\_CBLCODE [526](#)
    - MT\_CCODE [526](#)
    - MT\_FRTNCODE [526](#)
    - MT\_ICODE [526](#)
    - MT\_MASK [526](#)
    - MT\_OBJECT [526](#)

MT\_PCODE 526

available values for pipe

MT\_ANY 592

MT\_CBLCODE 592

MT\_CCODE 592

MT\_FRTNCODE 592

MT\_ICODE 592

MT\_MASK 592

MT\_OBJECT 592

MT\_PCODE 592

available values for RBF

MT\_ANY 460

MT\_CBLCODE 460

MT\_CCODE 460

MT\_FRTNCODE 460

MT\_ICODE 460

MT\_MASK 460

MT\_OBJECT 460

MT\_PCODE 460

available values for SBF

MT\_ANY 416

MT\_CBLCODE 416

MT\_CCODE 416

MT\_FRTNCODE 416

MT\_ICODE 416

MT\_MASK 416

MT\_OBJECT 416

MT\_PCODE 416

available values for SCF

ML\_ANY 186

ML\_CBLCODE 186

ML\_CCODE 186

ML\_FRTNCODE 186

ML\_ICODE 186

ML\_MASK 186

ML\_OBJECT 186

ML\_PCODE 186

line delete

changing 386

line feed character

changing 390, 392

## LINEDEL

changing  
for SCF 386

lines per page  
setting 218

lines per screen  
setting 218

LLPM\_COUNT  
changing 70

llpm\_count  
changing  
for cnfgdata 70

LLPM\_MAXCONNS  
changing 69

LLPM\_MAXPROTOS  
changing 67

LLPM\_MAXRCVMBUFS  
changing 68

logical sector offset  
changing 496, 562

logical unit static storage  
changing 192, 422, 466, 532, 598  
device identifier 192, 422, 466, 532, 598

lolim  
changing  
for init 162

LSNOFFS  
changing  
for PCF 562  
for RBF 496

lu\_ctrlrid  
changing  
for PCF 573  
for RBF 507  
PCF 574

lu\_lun  
changing  
for PCF 572  
for RBF 506

lu\_stp  
changing



- for PCF [569](#)
  - for RBF [503](#)
- lu\_tfm
  - changing
    - for PCF [571](#)
    - for RBF [505](#)
- lu\_totcyls
  - changing
    - RBF [508](#)
- LUN
  - changing
    - for PCF [532](#)
    - for pipe [598](#)
    - for RBF [466](#)
    - for SBF [422](#)
    - for SCF [192](#), [213](#)
- LUPARITY
  - changing
    - for SCF [229](#)

---

**M**

- m\_access
  - changing
    - for cnfgdata [21](#)
    - for init [113](#)
    - for PCF [522](#)
    - for pipe [588](#)
    - for RBF [456](#)
    - for SBF [412](#)
    - for SCF [182](#)
- m\_attrev (attributes/revision)
  - changing
    - for cnfgdata [26](#)
    - for init [118](#)
    - for PCF [527](#)
    - for pipe [593](#)
    - for RBF [461](#)
    - for SBF [417](#)
    - for SCF [187](#)
- m\_compat

- changing
    - for init [152](#)
- m\_cpucompat
  - changing
    - init [146](#)
- m\_cputyp
  - changing
    - for init [125](#)
- m\_dsptbl
  - changing
    - for init [145](#)
- m\_edit
  - changing
    - for cnfgdata [28](#)
    - for init [120](#)
    - for PCF [529](#)
    - for pipe [595](#)
    - for RBF [463](#)
    - for SBF [419](#)
    - for SCF [189](#)
- m\_edition
  - changing
    - for init [151](#)
- m\_events
  - changing
    - for init [139](#)
- m\_exec [108](#)
- m\_group
  - changing
    - for cnfgdata [18](#)
    - for init [110](#)
    - for PCF [519](#)
    - for pipe [585](#)
    - for RBF [453](#)
    - for SBF [409](#)
    - for SCF [179](#)
- m\_level
  - changing
    - for init [148](#)
- m\_major
  - changing

for init 149  
 m\_maxage  
   changing  
     for init 144  
 m\_maxsigs  
   changing  
     for init 154  
 m\_minor  
   changing  
     for init 150  
 m\_minpty  
   changing  
     for init 143  
 m\_paths  
   changing  
     for init 138  
 m\_procs  
   changing  
     for init 137  
 m\_site  
   changing  
     for init 124  
 m\_slice  
   changing  
     init 141  
 m\_syspri  
   changing  
     for init 142  
 m\_ticsec  
   changing  
     for init 140  
 m\_tmzone  
   changing  
     for init 147  
 m\_tylan  
   available values for cnfgdata  
     MT\_ANY 23  
     MT\_CDBDATA 24  
     MT\_DATA 24  
     MT\_DEVDESC 24  
     MT\_DEVDRVR 24

```

MT_FILEMAN 24
MT_MASK 24
MT_MULTI (reserved) 24
MT_PROGRAM 23
MT_SUBROUT 24
MT_SYSTEM 24
MT_TRAPLIB 24
available values for init
MT_ANY 116
MT_CDBDATA 116
MT_DATA 116
MT_DEVDESC 116
MT_DEVDRVR 116
MT_FILEMAN 116
MT_MASK 117
MT_MULTI (reserved) 116
MT_PROGRAM 116
MT_SUBROUT 116
MT_SYSTEM 116
MT_TRAPLIB 116
available values for PCF
MT_ANY 525
MT_CDBDATA 525
MT_DATA 525
MT_DEVDESC 525
MT_DEVDRVR 525
MT_FILEMAN 525
MT_MASK 526
MT_MULTI (reserved) 525
MT_PROGRAM 525
MT_SUBROUT 525
MT_SYSTEM 525
MT_TRAPLIB 525
available values for pipe
MT_ANY 591
MT_CDBDATA 591
MT_DATA 591
MT_DEVDESC 591
MT_DEVDRVR 591
MT_FILEMAN 591
MT_MASK 592

```

MT\_MULTI (reserved) 591  
 MT\_PROGRAM 591  
 MT\_SUBROUT 591  
 MT\_SYSTEM 591  
 MT\_TRAPLIB 591  
 available values for RBF  
 MT\_ANY 459  
 MT\_CDBDATA 459  
 MT\_DATA 459  
 MT\_DEVDESC 459  
 MT\_DEVDRVR 459  
 MT\_FILEMAN 459  
 MT\_MASK 460  
 MT\_MULTI (reserved) 459  
 MT\_PROGRAM 459  
 MT\_SUBROUT 459  
 MT\_SYSTEM 459  
 MT\_TRAPLIB 459  
 available values for SBF  
 MT\_ANY 415  
 MT\_CDBDATA 415  
 MT\_DATA 415  
 MT\_DEVDESC 415  
 MT\_DEVDRVR 415  
 MT\_FILEMAN 415  
 MT\_MASK 416  
 MT\_MULTI (reserved) 415  
 MT\_PROGRAM 415  
 MT\_SUBROUT 415  
 MT\_SYSTEM 415  
 MT\_TRAPLIB 415  
 available values for SCF  
 MT\_ANY 185  
 MT\_CDBDATA 185  
 MT\_DATA 185  
 MT\_DEVDESC 185  
 MT\_DEVDRVR 185  
 MT\_FILEMAN 185  
 MT\_MASK 186  
 MT\_MULTI (reserved) 185  
 MT\_PROGRAM 185

- MT\_SUBROUT 185
- MT\_SYSTEM 185
- MT\_TRAPLIB 185
- m\_tylan (type/language)
  - changing
    - for cnfgdata 23
    - for init 115
    - for PCF 524
    - for pipe 590
    - for RBF 458
    - for SBF 414
    - for SCF 184
- m\_user
  - changing
    - for cnfgdata 19
    - for init 111
    - for PCF 520
    - for pipe 586
    - for RBF 454
    - for SBF 410
    - for SCF 180
- MA\_GHOST
  - module attribute for cnfgdata
    - ghost 27
  - module attribute for init
    - ghost 119
  - module attribute for PCF
    - ghost 528
  - module attribute for pipe
    - ghost 594
  - module attribute for RBF
    - ghost 462
  - module attribute for SBF
    - ghost 418
  - module attribute for SCF
    - ghost 188
- MA\_MASK 418
- MA\_REENT
  - module attribute for cnfgdata
    - re-entrant 27
  - module attribute for init

- re-entrant [119](#)
  - module attribute for PCF
    - re-entrant [528](#)
  - module attribute for pipe
    - re-entrant [594](#)
  - module attribute for RBF
    - re-entrant [462](#)
  - module attribute for SBF
    - re-entrant [418](#)
  - module attribute for SCF
    - re-entrant [188](#)
- MA\_SUPER
  - module attribute for cnfgdata
    - system-state [27](#)
  - module attribute for init
    - system-state [119](#)
  - module attribute for PCF
    - system-state [528](#)
  - module attribute for pipe
    - system-state [594](#)
  - module attribute for RBF
    - system-state [462](#)
  - module attribute for SBF
    - system-state [418](#)
  - module attribute for SCF
    - system-state [188](#)
- mac\_address
  - changing
    - for cnfgdata [77](#)
  - interface data
    - configuration [77](#)
- mask
  - module header
    - type code for cnfgdata [24](#)
    - type code for init [117](#)
    - type code for PCF [526](#)
    - type code for pipe [592](#)
    - type code for RBF [460](#)
    - type code for SBF [416](#)
    - type code for SCF [186](#)
- MAX\_NOTIFIERS

- changing 97
- max\_notifiers
  - changing
    - for cnfgdata 97
- MAXBUFF
  - changing
    - for SCF 215
- maxllpmconns
  - changing
    - for cnfgdata 69
- maxllpmprotos
  - changing
    - for cnfgdata 67
- MAXPTY
  - changing
    - for init 144
- maxrcvmbufs
  - changing
    - for cnfgdata 68
- MAXSIGS
  - changing
    - for init 154
- MEM\_SHARED 157
- MEM\_SYS 157
- memory transfer size
  - changing 563
- memory.h 157
- MH\_ACCESS
  - changing
    - for cnfgdata 21
    - for init 113
    - for PCF 522
    - for pipe 588
    - for RBF 456
    - for SBF 412
    - for SCF 182
- MH\_ATTREV
  - changing
    - for cnfgdata 26
    - for init 118
    - for PCF 527



for pipe 593  
 for RBF 461  
 for SBF 417  
 for SCF 187

## MH\_EDITION

### changing

for cnfgdata 28  
 for init 120  
 for PCF 529  
 for pipe 595  
 for RBF 463  
 for SBF 419  
 for SCF 189

## MH\_GROUP

### changing

for cnfgdata 18  
 for init 110  
 for PCF 519  
 for pipe 585  
 for RBF 453  
 for SBF 409  
 for SCF 179

## MH\_NAME

### changing

for cnfgdata 20  
 for init 112  
 for PCF 521  
 for pipe 587  
 for RBF 455  
 for SBF 411  
 for SCF 181

## MH\_TYLAN

### changing

for cnfgdata 23  
 for init 115  
 for PCF 524  
 for pipe 590  
 for RBF 458  
 for SBF 414  
 for SCF 184

## MH\_USER

- changing
  - for cnfgdata 19
  - for init 111
  - for PCF 520
  - for pipe 586
  - for RBF 454
  - for SBF 410
  - for SCF 180

## MINPTY

- changing
  - for init 143

## ML\_ANY

- language code for SCF
  - wildcard value 186

## ML\_CBLCODE

- language code
  - COBOL I-code (reserved) 25, 117
- language code for PCF
  - COBOL I-code (reserved) 526
- language code for pipe
  - COBOL I-code (reserved) 592
- language code for RBF
  - COBOL I-code (reserved) 460
- language code for SBF
  - COBOL I-code (reserved) 416
- language code for SCF
  - COBOL I-code (reserved) 186

## ML\_CCODE

- language code
  - C I-code (reserved) 25, 117
- language code for PCF
  - C I-code (reserved) 526
- language code for pipe
  - C I-code (reserved) 592
- language code for RBF
  - C I-code (reserved) 460
- language code for SBF
  - C I-code (reserved) 416
- language code for SCF
  - C I-code (reserved) 186

## ML\_FRTNCODE

language code  
Fortran 25, 117  
language code for PCF  
Fortran 526  
language code for pipe  
Fortran 592  
language code for RBF  
Fortran 460  
language code for SBF  
Fortran 416  
language code for SCF  
Fortran 186

#### ML\_ICODE

language code for cnfgdata  
intermediate code 25  
language code for init  
intermediate code 117  
language code for PCF  
intermediate code 526  
language code for pipe  
intermediate code 592  
language code for RBF  
intermediate code 460  
language code for SBF  
intermediate code 416  
language code for SCF  
intermediate code 186

#### ML\_MASK

language code  
mask 25, 117  
language code for PCF  
mask 526  
language code for pipe  
mask 592  
language code for RBF  
mask 460  
language code for SBF  
mask 416  
language code for SCF  
mask 186

#### ML\_OBJECT

language code for cnfgdata  
     machine language [25](#)  
 language code for init  
     machine language [117](#)  
 language code for PCF  
     machine language [526](#)  
 language code for pipe  
     machine language [592](#)  
 language code for RBF  
     machine language [460](#)  
 language code for SBF  
     machine language [416](#)  
 language code for SCF  
     machine language [186](#)

#### ML\_PCODE

language code for cnfgdata  
     PASCAL [25](#)  
 language code for init  
     PASCAL [117](#)  
 language code for PCF  
     PASCAL [526](#)  
 language code for pipe  
     PASCAL [592](#)  
 language code for RBF  
     PASCAL [460](#)  
 language code for SBF  
     PASCAL [416](#)  
 language code for SCF  
     PASCAL [186](#)

#### mod\_name

changing  
     for cnfgdata [20](#)  
     for init [112](#)  
     PCF descriptor name [521](#)  
     pipe descriptor name [587](#)  
     RBF descriptor name [455](#)  
     SBF descriptor name [411](#)  
     SCF descriptor name [181](#)

#### modes.h

    setting dd\_mode [198](#), [427](#), [471](#), [537](#), [603](#)

#### module

- header [108](#)
- module header
  - attributes/revision
    - setting for cnfgdata [26](#)
    - setting for init [118](#)
    - setting for PCF [527](#)
    - setting for pipe [593](#)
    - setting for RBF [461](#)
    - setting for SBF [417](#)
    - setting for SCF [187](#)
  - cnfgdata
    - access permissions
      - setting [21](#)
  - edition number
    - setting for cnfgdata [28](#)
    - setting for init [120](#)
    - setting for PCF [529](#)
    - setting for pipe [595](#)
    - setting for RBF [463](#)
    - setting for SBF [419](#)
    - setting for SCF [189](#)
  - init
    - access permissions
      - setting [113](#)
  - PCF
    - access permissions
      - setting [522](#)
  - pipe
    - access permissions
      - setting [588](#)
  - RBF
    - access permissions
      - setting [456](#)
  - SBF
    - access permissions
      - setting [412](#)
  - SCF
    - access permissions
      - setting [182](#)
  - type and language
    - setting for cnfgdata [23](#)

- setting for init [115](#)
  - setting for PCF [524](#)
  - setting for pipe [590](#)
  - setting for RBF [458](#)
  - setting for SBF [414](#)
  - setting for SCF [184](#)
- user ID
  - cnfgdata [19](#)
  - init [111](#)
  - PCF [520](#)
  - pipe [586](#)
  - RBF [454](#)
  - SBF [410](#)
  - SCF [180](#)
- module.h
  - access permissions
    - available values for cnfgdata [21](#), [113](#)
    - available values for PCF [522](#)
    - available values for pipe [588](#)
    - available values for RBF [456](#)
    - available values for SBF [412](#)
    - available values for SCF [182](#)
  - attribute/revision
    - available values for cnfgdata [26](#)
    - available values for init [118](#)
    - available values for PCF [527](#)
    - available values for pipe [593](#)
    - available values for RBF [461](#)
    - available values for SBF [417](#)
    - available values for SCF [187](#)
  - type/language
    - available values for cnfgdata [23](#)
    - available values for init [116](#)
    - available values for PCF [525](#)
    - available values for pipe [591](#)
    - available values for RBF [459](#)
    - available values for SBF [415](#)
    - available values for SCF [185](#)
- MP\_GROUP\_EXEC
  - setting module permissions
    - for cnfgdata [22](#)

- for init [114](#)
  - for PCF [523](#)
  - for pipe [589](#)
  - for RBF [457](#)
  - for SBF [413](#)
  - for SCF [183](#)
- MP\_GROUP\_MASK
  - setting module permissions
    - for cnfgdata [22](#)
  - for init [114](#)
  - for PCF [523](#)
  - for pipe [589](#)
  - for RBF [457](#)
  - for SBF [413](#)
  - for SCF [183](#)
- MP\_GROUP\_READ
  - setting module permissions
    - for cnfgdata [22](#)
  - for init [114](#)
  - for PCF [523](#)
  - for pipe [589](#)
  - for RBF [457](#)
  - for SBF [413](#)
  - for SCF [183](#)
- MP\_GROUP\_WRITE
  - setting module permissions
    - for cnfgdata [22](#)
  - for init [114](#)
  - for PCF [523](#)
  - for pipe [589](#)
  - for RBF [457](#)
  - for SBF [413](#)
  - for SCF [183](#)
- MP\_OWNER\_EXEC
  - setting module permissions
    - for cnfgdata [21](#)
  - for init [114](#)
  - for PCF [523](#)
  - for pipe [589](#)
  - for RBF [457](#)
  - for SBF [413](#)

```

        for SCF 183
MP_OWNER_MASK
    setting module permissions
        for cnfgdata 22
        for init 114
        for PCF 523
        for pipe 589
        for RBF 457
        for SBF 413
        for SCF 183
MP_OWNER_READ
    setting module permissions
        for cnfgdata 21
        for init 113
        for PCF 522
        for pipe 589
        for RBF 456
        for SBF 412
        for SCF 182
MP_OWNER_WRITE
    setting module permissions
        for cnfgdata 21
        for init 113
        for PCF 522
        for pipe 589
        for RBF 456
        for SBF 412
        for SCF 182
MP_SYSTM_MASK
    setting module permissions
        for cnfgdata 22
        for init 114
        for PCF 523
        for pipe 589
        for RBF 457
        for SBF 413
        for SCF 183
MP_WORLD_ACCESS
    setting module permissions
        for cnfgdata 22
        for init 114

```



- for PCF 523
- for pipe 589
- for RBF 457
- for SBF 413
- for SCF 183

#### MP\_WORLD\_EXEC

- setting module permissions

- for cnfgdata 22
- for init 114
- for PCF 523
- for pipe 589
- for RBF 457
- for SBF 413
- for SCF 183

#### MP\_WORLD\_MASK

- setting module permissions

- for cnfgdata 22
- for init 114
- for PCF 523
- for pipe 589
- for RBF 457
- for SBF 413
- for SCF 183

#### MP\_WORLD\_READ

- setting module permissions

- for cnfgdata 22
- for init 114
- for PCF 523
- for pipe 589
- for RBF 457
- for SBF 413
- for SCF 183

#### MP\_WORLD\_WRITE

- setting module permissions

- for cnfgdata 22
- for init 114
- for PCF 523
- for pipe 589
- for RBF 457
- for SBF 413
- for SCF 183

## MPUCHIP

changing

for init 125

MR\_MASK 418

## MT\_ANY

language code for cnfgdata

wildcard value 25

language code for init

wildcard value 117

language code for PCF

wildcard value 526

language code for pipe

wildcard value 592

language code for RBF

wildcard value 460

language code for SBF

wildcard value 416

m\_tylan field for cnfgdata

wildcard value 23

m\_tylan field for init

wildcard value 116

m\_tylan field for PCF

wildcard value 525

m\_tylan field for pipe

wildcard value 591

m\_tylan field for RBF

wildcard value 459

m\_tylan field for SBF

wildcard value 415

m\_tylan field for SCF

wildcard value 185

## MT\_CDBDATA

m\_tylan field for cnfgdata

configuration data block value 24

m\_tylan field for init

configuration data block value 116

m\_tylan field for PCF

configuration data block value 525

m\_tylan field for pipe

configuration data block value 591

m\_tylan field for RBF

|                                |     |
|--------------------------------|-----|
| configuration data block value | 459 |
| m_tylan field for SBF          |     |
| configuration data block value | 415 |
| m_tylan field for SCF          |     |
| configuration data block value | 185 |
| MT_DATA                        |     |
| m_tylan field for cnfgdata     |     |
| data value                     | 24  |
| m_tylan field for init         |     |
| data value                     | 116 |
| m_tylan field for PCF          |     |
| data value                     | 525 |
| m_tylan field for pipe         |     |
| data value                     | 591 |
| m_tylan field for RBF          |     |
| data value                     | 459 |
| m_tylan field for SBF          |     |
| data value                     | 415 |
| m_tylan field for SCF          |     |
| data value                     | 185 |
| MT_DEVDESC                     |     |
| m_tylan field for cnfgdata     |     |
| device descriptor value        | 24  |
| m_tylan field for init         |     |
| device descriptor value        | 116 |
| m_tylan field for PCF          |     |
| device descriptor value        | 525 |
| m_tylan field for pipe         |     |
| device descriptor value        | 591 |
| m_tylan field for RBF          |     |
| device descriptor value        | 459 |
| m_tylan field for SBF          |     |
| device descriptor value        | 415 |
| m_tylan field for SCF          |     |
| device descriptor value        | 185 |
| MT_DEVDRVR                     |     |
| m_tylan field for cnfgdata     |     |
| physical device driver value   | 24  |
| m_tylan field for init         |     |
| physical device driver value   | 116 |
| m_tylan field for PCF          |     |

|                              |     |
|------------------------------|-----|
| physical device driver value | 525 |
| m_tylan field for pipe       |     |
| physical device driver value | 591 |
| m_tylan field for RBF        |     |
| physical device driver value | 459 |
| m_tylan field for SBF        |     |
| physical device driver value | 415 |
| m_tylan field for SCF        |     |
| physical device driver value | 185 |
| MT_FILEMAN                   |     |
| m_tylan field for cnfgdata   |     |
| file manager value           | 24  |
| m_tylan field for init       |     |
| file manager value           | 116 |
| m_tylan field for PCF        |     |
| file manager value           | 525 |
| m_tylan field for pipe       |     |
| file manager value           | 591 |
| m_tylan field for RBF        |     |
| file manager value           | 459 |
| m_tylan field for SBF        |     |
| file manager value           | 415 |
| m_tylan field for SCF        |     |
| file manager value           | 185 |
| MT_MASK                      |     |
| m_tylan field for cnfgdata   | 24  |
| m_tylan field for init       | 117 |
| m_tylan field for PCF        | 526 |
| m_tylan field for pipe       | 592 |
| m_tylan field for RBF        | 460 |
| m_tylan field for SBF        | 416 |
| m_tylan field for SCF        | 186 |
| MT_MULTI                     |     |
| m_tylan field for cnfgdata   |     |
| reserved value               | 24  |
| m_tylan field for init       |     |
| reserved value               | 116 |
| m_tylan field for PCF        |     |
| reserved value               | 525 |
| m_tylan field for pipe       |     |
| reserved value               | 591 |

m\_tylan field for RBF  
     reserved value [459](#)  
 m\_tylan field for SBF  
     reserved value [415](#)  
 m\_tylan field for SCF  
     reserved value [185](#)

#### MT\_PROGRAM

m\_tylan field for cnfgdata  
     executable value [23](#)  
 m\_tylan field for init  
     executable value [116](#)  
 m\_tylan field for PCF  
     executable value [525](#)  
 m\_tylan field for pipe  
     executable value [591](#)  
 m\_tylan field for RBF  
     executable value [459](#)  
 m\_tylan field for SBF  
     executable value [415](#)  
 m\_tylan field for SCF  
     executable value [185](#)

#### MT\_SUBROUT

m\_tylan field for cnfgdata  
     subroutine value [24](#)  
 m\_tylan field for init  
     subroutine value [116](#)  
 m\_tylan field for PCF  
     subroutine value [525](#)  
 m\_tylan field for pipe  
     subroutine value [591](#)  
 m\_tylan field for RBF  
     subroutine value [459](#)  
 m\_tylan field for SBF  
     subroutine value [415](#)  
 m\_tylan field for SCF  
     subroutine value [185](#)

#### MT\_SYSTEM [107](#)

m\_tylan field for cnfgdata  
     system module value [24](#)  
 m\_tylan field for init  
     system module value [116](#)

- m\_tylan field for PCF
    - system module value 525
  - m\_tylan field for pipe
    - system module value 591
  - m\_tylan field for RBF
    - system module value 459
  - m\_tylan field for SBF
    - system module value 415
  - m\_tylan field for SCF
    - system module value 185
- MT\_TRAPLIB
  - m\_tylan field for cnfgdata
    - trap library value 24
  - m\_tylan field for init
    - trap library value 116
  - m\_tylan field for PCF
    - trap library value 525
  - m\_tylan field for pipe
    - trap library value 591
  - m\_tylan field for RBF
    - trap library value 459
  - m\_tylan field for SBF
    - trap library value 415
  - m\_tylan field for SCF
    - trap library value 185

N

- name string
  - device driver
    - selecting for PCF 540
    - selecting for pipe 606
    - selecting for RBF 474
    - selecting for SBF 430
    - selecting for SCF 201
  - file manager
    - selecting for PCF 539
    - selecting for pipe 605
    - selecting for RBF 473
    - selecting for SBF 429
    - selecting for SCF 200

---

**O**

object code  
    module header  
        language code for cnfgdata 25  
        language code for init 117  
        language code for PCF 526  
        language code for pipe 592  
        language code for RBF 460  
        language code for SBF 416  
        language code for SCF 186  
OS\_EDITION  
    changing  
        for init 151  
OS\_LEVEL  
    changing  
        for init 148  
OS\_REVISION  
    changing 150  
OS\_VERSION  
    changing  
        for init 149  
OS9K\_REVSTR  
    changing  
        for init 127  
os9rev\_name  
    changing  
        for init 127  
OUTPUT\_TYPE  
    changing  
        for SCF 212  
OUTSIZE  
    changing  
        for SCF 217

---

**P**

PAGE\_SIZE  
    changing  
        for SCF 218  
PAGEPAUSE

- changing
    - for SCF [392](#)
- PAGESIZE
  - changing
    - for SCF [397](#)
- PARK
  - changing
    - for PCF [561](#)
    - for RBF [495](#)
- parking
  - disk heads,
    - changing cylinder location for [495](#), [561](#)
- PASCAL
  - module header
    - language code for cnfgdata [25](#)
    - language code for init [117](#)
    - language code for PCF [526](#)
    - language code for pipe [592](#)
    - language code for RBF [460](#)
    - language code for SBF [416](#)
    - language code for SCF [186](#)
- path descriptor
  - size
    - dd\_pd\_size [193](#), [423](#), [467](#), [533](#), [599](#)
    - for IOMAN [193](#), [423](#), [467](#), [533](#), [599](#)
- PATHS
  - changing
    - for init [138](#)
- pause control key
  - changing for keyboard [223](#)
- PCF
  - BLKSIZE
    - changing [556](#)
  - changing
    - BLKOFFS [554](#)
    - BLKSTRK [549](#)
    - BLKSTRK0 [550](#)
    - CONTROL [557](#)
    - CTRLRID [573](#), [574](#)
    - CYLNDRS [548](#)
    - dd\_class field [541](#)



dd\_lun field [532](#)  
 dd\_mode field [536](#)  
 dd\_pd\_size field [533](#)  
 dd\_port field [531](#)  
 dd\_type field [534](#)  
 drvr\_name field [540](#)  
 fmgr\_name field [539](#)  
 FORMAT [546](#)  
 INTRLV [552](#)  
 IRQLEVEL [566](#)  
 LSNOFFS [562](#)  
 lu\_ctrlrid field [573](#), [574](#)  
 lu\_lun field [572](#)  
 lu\_stp field [569](#)  
 lu\_tfm field [571](#)  
 m\_access field [522](#)  
 m\_attrev (attributes/revision) field [527](#)  
 m\_edit field [529](#)  
 m\_group field [519](#)  
 m\_tylan (type/language) field [524](#)  
 m\_user field [520](#)  
 PARK [561](#)  
 pd\_blk field [549](#)  
 pd\_boffs field [554](#)  
 pd\_bsize field [556](#)  
 pd\_cntl field [557](#)  
 pd\_cyl field [548](#)  
 pd\_format field [546](#)  
 pd\_ilv field [552](#)  
 pd\_lsnoffs field [562](#)  
 pd\_park field [561](#)  
 pd\_rwr field [560](#)  
 pd\_sas field [551](#)  
 pd\_sid field [544](#)  
 pd\_t0b field [550](#)  
 pd\_toffs field [553](#)  
 pd\_trys field [555](#)  
 pd\_vfy field [545](#)  
 pd\_wpc field [559](#)  
 pd\_xfersize field [563](#)  
 PRECOMP [559](#)

- REDWRITE 560
- SCSILUN 572
- SEGSIZE 551
- STEP 569
- TRKOFFS 553
- TRYS 555
- v\_irqlevel field 566
- v\_priority field 567
- v\_vector field 565
- VECTOR 565
- XFERSIZE 563
- DD\_CLASS
  - changing 541
- DD\_MODE
  - changing 536
- DD\_TYPE
  - changing 534
- descriptor name
  - changing mod\_name field 521
- device driver
  - port address offset 531
- DMAMODE
  - changing 571
- DRVR\_NAME
  - changing 540
- FMGR\_NAME
  - changing 539
- LUN
  - changing 532
- MH\_ACCESS
  - changing 522
- MH\_ATTREV
  - changing 527
- MH\_EDITION
  - changing 529
- MH\_GROUP
  - changing 519
- MH\_NAME
  - changing 521
- MH\_TYLAN
  - changing 524

MH\_USER  
     changing 520  
 module header  
     group ID 519  
 PD\_SIZE  
     changing 533  
 PORTADDR  
     changing 531  
 user ID  
     module header 520  
 pd\_alf  
     changing  
         for SCF 390  
 pd\_backsp  
     changing  
         for SCF 384  
 pd\_bellch  
     changing  
         for SCF 380  
 pd\_blk  
     changing  
         for PCF 549  
         for RBF 483  
 pd\_blksiz  
     changing  
         for SBF 433  
 pd\_boffs  
     changing  
         for PCF 554  
         for RBF 488  
 pd\_bsize  
     changing  
         for PCF 556  
         for RBF 490  
 pd\_bspch  
     changing  
         for SCF 381  
 pd\_case  
     changing  
         for SCF 382  
 pd\_cntl

- changing
    - for PCF [557](#)
    - for RBF [491](#)
- pd\_cyl
  - changing
    - for PCF [548](#)
    - for RBF [482](#)
- pd\_delete
  - changing
    - for SCF [386](#)
- pd\_dmamode
  - changing
    - for SBF [435](#)
- pd\_echo
  - changing
    - for SCF [388](#)
- pd\_eofch
  - changing
    - for SCF [378](#)
- pd\_eorch
  - changing
    - for SCF [375](#)
- pd\_flags
  - changing
    - for SBF [434](#)
- pd\_format
  - changing
    - for PCF [546](#)
    - for RBF [480](#)
- pd\_ilv
  - changing
    - for PCF [552](#)
    - for RBF [486](#)
- pd\_inmap0func\_code
  - changing
    - for SCF [247](#)
- pd\_inmap0size
  - changing
    - for SCF [249](#)
- pd\_inmap0string
  - changing

```

        for SCF 250
pd_inmap0type
    changing
        for SCF 245
pd_inmap10func_code
    changing
        for SCF 288
pd_inmap10size
    changing
        for SCF 289
pd_inmap10string
    changing
        for SCF 290
pd_inmap10type
    changing
        for SCF 287
pd_inmap11func_code
    changing
        for SCF 292
pd_inmap11size
    changing
        for SCF 293
pd_inmap11string
    changing
        for SCF 294
pd_inmap11type
    changing
        for SCF 291
pd_inmap12func_code
    changing
        for SCF 296
pd_inmap12size
    changing
        for SCF 297
pd_inmap12string
    changing
        for SCF 298
pd_inmap12type
    changing
        for SCF 295
pd_inmap13func_code

```

changing  
        for SCF [300](#)  
pd\_inmap13size  
    changing  
        for SCF [301](#)  
pd\_inmap13string  
    changing  
        for SCF [302](#)  
pd\_inmap13type  
    changing  
        for SCF [299](#)  
pd\_inmap14func\_code  
    changing  
        for SCF [304](#)  
pd\_inmap14size  
    changing  
        for SCF [305](#)  
pd\_inmap14string  
    changing  
        for SCF [306](#)  
pd\_inmap14type  
    changing  
        for SCF [303](#)  
pd\_inmap15func\_code  
    changing  
        for SCF [308](#)  
pd\_inmap15size  
    changing  
        for SCF [309](#)  
pd\_inmap15string  
    changing  
        for SCF [310](#)  
pd\_inmap15type  
    changing  
        for SCF [307](#)  
pd\_inmap16func\_code  
    changing  
        for SCF [312](#)  
pd\_inmap16size  
    changing  
        for SCF [313](#)

pd\_inmap16string  
    changing  
        for SCF 314

pd\_inmap16type  
    changing  
        for SCF 311

pd\_inmap17func\_code  
    changing  
        for SCF 316

pd\_inmap17size  
    changing  
        for SCF 317

pd\_inmap17string  
    changing  
        for SCF 318

pd\_inmap17type  
    changing  
        for SCF 315

pd\_inmap18func\_code  
    changing  
        for SCF 320

pd\_inmap18size  
    changing  
        for SCF 321

pd\_inmap18string  
    changing  
        for SCF 322

pd\_inmap18type  
    changing  
        for SCF 319

pd\_inmap19func\_code  
    changing  
        for SCF 324

pd\_inmap19size  
    changing  
        for SCF 325

pd\_inmap19string  
    changing  
        for SCF 326

pd\_inmap19type  
    changing

```

        for SCF 323
pd_inmap1func_code
    changing
        for SCF 252
pd_inmap1size
    changing
        for SCF 253
pd_inmap1string
    changing
        for SCF 254
pd_inmap1type
    changing
        for SCF 251
pd_inmap20func_code
    changing
        for SCF 328
pd_inmap20size
    changing
        for SCF 329
pd_inmap20string
    changing
        for SCF 330
pd_inmap20type
    changing
        for SCF 327
pd_inmap21func_code
    changing
        for SCF 332
pd_inmap21size
    changing
        for SCF 333
pd_inmap21string
    changing
        for SCF 334
pd_inmap21type
    changing
        for SCF 331
pd_inmap22func_code
    changing
        for SCF 336
pd_inmap22size

```



changing  
for SCF 337

pd\_inmap22string  
changing  
for SCF 338

pd\_inmap22type  
changing  
for SCF 335

pd\_inmap23func\_code  
changing  
for SCF 340

pd\_inmap23size  
changing  
for SCF 341

pd\_inmap23string  
changing  
for SCF 342

pd\_inmap23type  
changing  
for SCF 339

pd\_inmap24func\_code  
changing  
for SCF 344

pd\_inmap24size  
changing  
for SCF 345

pd\_inmap24string  
changing  
for SCF 346

pd\_inmap24type  
changing  
for SCF 343

pd\_inmap25func\_code  
changing  
for SCF 348

pd\_inmap25size  
changing  
for SCF 349

pd\_inmap25string  
changing  
for SCF 350

pd\_inmap25type  
    changing  
        for SCF [347](#)

pd\_inmap26func\_code  
    changing  
        for SCF [352](#)

pd\_inmap26size  
    changing  
        for SCF [353](#)

pd\_inmap26string  
    changing  
        for SCF [354](#)

pd\_inmap26type  
    changing  
        for SCF [351](#)

pd\_inmap27func\_code  
    changing  
        for SCF [356](#)

pd\_inmap27size  
    changing  
        for SCF [357](#)

pd\_inmap27string  
    changing  
        for SCF [358](#)

pd\_inmap27type  
    changing  
        for SCF [355](#)

pd\_inmap28func\_code  
    changing  
        for SCF [360](#)

pd\_inmap28size  
    changing  
        for SCF [361](#)

pd\_inmap28string  
    changing  
        for SCF [362](#)

pd\_inmap28type  
    changing  
        for SCF [359](#)

pd\_inmap29func\_code  
    changing

for SCF [364](#)  
pd\_inmap29size  
changing  
for SCF [365](#)  
pd\_inmap29string  
changing  
for SCF [366](#)  
pd\_inmap29type  
changing  
for SCF [363](#)  
pd\_inmap2func\_code  
changing  
for SCF [256](#)  
pd\_inmap2size  
changing  
for SCF [257](#)  
pd\_inmap2string  
changing  
for SCF [258](#)  
pd\_inmap2type  
changing  
for SCF [255](#)  
pd\_inmap30func\_code  
changing  
for SCF [368](#)  
pd\_inmap30size  
changing  
for SCF [369](#)  
pd\_inmap30string  
changing  
for SCF [370](#)  
pd\_inmap30type  
changing  
for SCF [367](#)  
pd\_inmap31func\_code  
changing  
for SCF [372](#)  
pd\_inmap31size  
changing  
for SCF [373](#)  
pd\_inmap31string

changing  
        for SCF [374](#)

pd\_inmap31type  
    changing  
        for SCF [371](#)

pd\_inmap3func\_code  
    changing  
        for SCF [260](#)

pd\_inmap3size  
    changing  
        for SCF [261](#)

pd\_inmap3string  
    changing  
        for SCF [262](#)

pd\_inmap3type  
    changing  
        for SCF [259](#)

pd\_inmap4func\_code  
    changing  
        for SCF [264](#)

pd\_inmap4size  
    changing  
        for SCF [265](#)

pd\_inmap4string  
    changing  
        for SCF [266](#)

pd\_inmap4type  
    changing  
        for SCF [263](#)

pd\_inmap5func\_code  
    changing  
        for SCF [268](#)

pd\_inmap5size  
    changing  
        for SCF [269](#)

pd\_inmap5string  
    changing  
        for SCF [270](#), [274](#)

pd\_inmap5type  
    changing  
        for SCF [267](#)

pd\_inmap6func\_code  
    changing  
        for SCF [272](#)

pd\_inmap6size  
    changing  
        for SCF [273](#)

pd\_inmap6type  
    changing  
        for SCF [271](#)

pd\_inmap7func\_code  
    changing  
        for SCF [276](#)

pd\_inmap7size  
    changing  
        for SCF [277](#)

pd\_inmap7string  
    changing  
        for SCF [278](#)

pd\_inmap7type  
    changing  
        for SCF [275](#)

pd\_inmap8func\_code  
    changing  
        for SCF [280](#)

pd\_inmap8size  
    changing  
        for SCF [281](#)

pd\_inmap8string  
    changing  
        for SCF [282](#)

pd\_inmap8type  
    changing  
        for SCF [279](#)

pd\_inmap9func\_code  
    changing  
        for SCF [284](#)

pd\_inmap9size  
    changing  
        for SCF [285](#)

pd\_inmap9string  
    changing

for SCF [286](#)  
pd\_inmap9type  
changing  
for SCF [283](#)  
pd\_insm  
changing  
for SCF [394](#)  
pd\_ksnoffs  
changing  
for PCF [562](#)  
for RBF [496](#)  
pd\_nulls  
changing  
for SCF [396](#)  
pd\_page  
changing  
for SCF [397](#)  
pd\_park  
changing  
PCF [561](#)  
RBF [495](#)  
pd\_pause  
changing  
for SCF [392](#)  
pd\_rwr  
changing  
PCF [560](#)  
RBF [494](#)  
pd\_sas  
changing  
PCF [551](#)  
RBF [485](#)  
pd\_sci\_id  
changing  
SBF [436](#)  
pd\_scsilun  
changing  
SBF [437](#)  
pd\_sid  
changing  
for PCF [544](#)

```

        for RBF 478
PD_SIZE
    changing
        for PCF 533
        for pipe 599
        for RBF 467
        for SBF 423
        for SCF 193
pd_t0b
    changing
        PCF 550
        RBF 484
pd_tabch
    changing
        for SCF 379
pd_tabsiz
    changing
        for SCF 398
pd_toffs
    changing
        PCF 553
        RBF 487
pd_trys
    changing
        PCF 555
        RBF 489
pd_vfy
    changing
        for PCF 545
        for RBF 479
pd_wpc
    changing
        for PCF 559
        for RBF 493
pd_xfersize
    changing
        for PCF 563
        for RBF 497
pipe
    changing
        BUFSZ 609

```

- bufsz field 609
- dd\_class field 607
- dd\_lun field 598
- dd\_mode field 602
- dd\_pd\_size field 599
- dd\_port field 597
- dd\_type field 600
- drv\_name field 606
- fmgr\_name field 605
- m\_access field 588
- m\_attrev (attributes/revision) field 593
- m\_edit field 595
- m\_group field 585
- m\_tylan (type/language) field 590
- m\_user field 586
- DD\_CLASS
  - changing 607
- DD\_MODE
  - changing 602
- DD\_TYPE
  - changing 600
- descriptor name
  - changing mod\_name field 587
- device driver
  - port address offset 597
- DRV\_NAME
  - changing 606
- FMGR\_NAME
  - changing 605
- LUN
  - changing 598
- MH\_ACCESS
  - changing 588
- MH\_ATTREV
  - changing 593
- MH\_EDITION
  - changing 595
- MH\_GROUP
  - changing 585
- MH\_NAME
  - changing 587



- MH\_TYLAN
  - changing [590](#)
- MH\_USER
  - changing [586](#)
- module header
  - group ID [585](#)
- PD\_SIZE
  - changing [599](#)
- PORTADDR
  - changing [597](#)
  - user ID
    - module header [586](#)
- polling interrupt
  - changing [441](#), [501](#), [567](#)
- port address
  - changing, for
    - PCF device [531](#)
    - pipe device [597](#)
    - RBF device [465](#)
    - SBF device [421](#)
    - SCF device [191](#)
- port\_address
  - changing
    - for cnfgdata [81](#)
  - interface data
    - configuration [81](#)
- PORTADDR
  - changing
    - for PCF [531](#)
    - for pipe [597](#)
    - for RBF [465](#)
    - for SBF [421](#)
    - for SCF [191](#)
- PRECOMP
  - changing
    - for PCF [559](#)
    - for RBF [493](#)
- precompensation
  - starting point for writing
    - changing [493](#), [559](#)
- preio\_name

changing  
for init 155

PREIOS  
changing  
for init 155

prior  
changing  
for init 158

PRIORITY  
changing 441, 501, 567  
for SCF 209

PROCS  
changing  
for init 137

---

Q

quit control key  
changing for keyboard 222

---

R

RBF

BLKSIZE  
changing 490

changing

BLKOFFS 488

BLKSTRK 483

BLKSTRK0 484

CONTROL 491

CTRLRID 507

CYLNDRS 482

dd\_class field 475

dd\_lun field 466

dd\_mode field 470

dd\_pd\_size field 467

dd\_port field 465

dd\_type field 468

drv\_name field 474

fmgr\_name field 473

FORMAT 480  
 INTRLV 486  
 IRQLEVEL 500  
 LSNOFFS 496  
 lu\_ctrlrid field 507  
 lu\_lun field 506  
 lu\_stp field 503  
 lu\_tfm field 505  
 lu\_totcyls field 508  
 m\_access field 456  
 m\_attrev (attributes/revision) field 461  
 m\_edit field 463  
 m\_group field 453  
 m\_tylan (type/language) field 458  
 m\_user field 454  
 PARK 495  
 pd\_blk field 483  
 pd\_boffs field 488  
 pd\_bsize field 490  
 pd\_cntl field 491  
 pd\_cyl field 482  
 pd\_format field 480  
 pd\_ilv field 486  
 pd\_lsnoffs field 496  
 pd\_park field 495  
 pd\_rwr field 494  
 pd\_sas field 485  
 pd\_sid field 478  
 pd\_t0b field 484  
 pd\_toffs field 487  
 pd\_trys field 489  
 pd\_vfy field 479  
 pd\_wpc field 493  
 pd\_xfersize field 497  
 PRECOMP 493  
 REDWRITE 494  
 SCSILUN 506  
 SEGSIZE 485  
 STEP 503  
 TOTCYLS 508  
 TRKOFFS 487

- TRYS [489](#)
- v\_irqlevel field [500](#)
- v\_priority field [501](#)
- v\_vector field [499](#)
- VECTOR [499](#)
- XFERSIZE [497](#)
- DD\_CLASS
  - changing [475](#)
- DD\_MODE
  - changing [470](#)
- DD\_TYPE
  - changing [468](#)
- descriptor name
  - changing mod\_name field [455](#)
- device driver
  - port address offset [465](#)
- DMAMODE
  - changing [505](#)
- DRVR\_NAME
  - changing [474](#)
- FMGR\_NAME
  - changing [473](#)
- LUN
  - changing [466](#)
- MH\_ACCESS
  - changing [456](#)
- MH\_ATTREV
  - changing [461](#)
- MH\_EDITION
  - changing [463](#)
- MH\_GROUP
  - changing [453](#)
- MH\_NAME
  - changing [455](#)
- MH\_TYLAN
  - changing [458](#)
- MH\_USER
  - changing [454](#)
- module header
  - group ID [453](#)
- PD\_SIZE

- changing 467
- PORTADDR
  - changing 465
- user ID
  - module header 454
- reduced write cylinder
  - starting point
    - changing 494, 560
- REDWRITE
  - changing
    - for PCF 560
    - for RBF 494
- re-entrant
  - module
    - setting value for cnfgdata 27
    - setting value for init 119
    - setting value for PCF 528
    - setting value for pipe 594
    - setting value for RBF 462
    - setting value for SBF 418
    - setting value for SCF 188
- request to send flag
  - changing for SCF device 234
- revision
  - of module
    - setting for cnfgdata 26
    - setting for init 118
    - setting for PCF 527
    - setting for pipe 593
    - setting for RBF 461
    - setting for SBF 417
    - setting for SCF 187
- RTC\_NAME
  - changing
    - for init 134
- rtc\_name
  - changing
    - for init 134
- RTSSTATE
  - changing
    - for SCF 234

---

**S**

|                 |      |      |      |      |     |
|-----------------|------|------|------|------|-----|
| S_IAPPEND       |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 198, | 427, | 471, | 537, | 603 |
| S_ICONTIG       |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 198, | 427, | 471, | 537, | 603 |
| S_ICREAT        |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 198, | 427, | 471, | 537, | 603 |
| S_IEXCL         |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 198, | 427, | 471, | 537, | 603 |
| S_IEXEC         |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 198, | 427, | 471, | 537, | 603 |
| S_IGEXEC        |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 199, | 428, | 472, | 538, | 604 |
| S_IGREAD        |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 199, | 428, | 472, | 538, | 604 |
| S_IGSEARCH      |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 199, | 428, | 472, | 538, | 604 |
| S_IGWRITE       |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 199, | 428, | 472, | 538, | 604 |
| S_IOEXEC        |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 199, | 428, | 472, | 538, | 604 |
| S_IOREAD        |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 199, | 428, | 472, | 538, | 604 |
| S_IOSEARCH      |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 199, | 428, | 472, | 538, | 604 |
| S_IOWRITE       |      |      |      |      |     |
| dd_mode         |      |      |      |      |     |
| available value | 199, | 428, | 472, | 538, | 604 |

S\_IPRM  
     dd\_mode  
         available value [198](#), [427](#), [471](#), [537](#), [603](#)  
 S\_IREAD  
     dd\_mode  
         available value [198](#), [427](#), [471](#), [537](#), [603](#)  
 S\_ISEARCH  
     dd\_mode  
         available value [198](#), [427](#), [471](#), [537](#), [604](#)  
 S\_ISHARE  
     dd\_mode  
         available value [198](#), [427](#), [471](#), [537](#), [603](#)  
 S\_ITRUNC  
     dd\_mode  
         available value [198](#), [427](#), [471](#), [537](#), [603](#)  
 S\_IWRITE  
     dd\_mode  
         available value [198](#), [427](#), [471](#), [537](#), [603](#)  
 SBF  
     changing  
         dd\_class field [431](#)  
         dd\_lun field [422](#)  
         dd\_mode field [426](#)  
         dd\_pd\_size field [423](#)  
         dd\_port field [421](#)  
         dd\_type field [424](#)  
         DMAMODE [435](#)  
         drv\_name field [430](#)  
         FLAGS [434](#)  
         fmgr\_name field [429](#)  
         IRQLEVEL [440](#)  
         m\_access field [412](#)  
         m\_attr (attributes/revision) field [417](#)  
         m\_edit field [419](#)  
         m\_group field [409](#)  
         m\_tylan (type/language) field [414](#)  
         m\_user field [410](#)  
         pd\_blksiz field [433](#)  
         pd\_dmamode field [435](#)  
         pd\_flags field [434](#)  
         pd\_sci\_id field [436](#)

- pd\_scsilun field [437](#)
- sbfdflag field [442](#)
- sbfdirqlevel field [440](#)
- sbfdpriority field [441](#)
- sbfdvector field [439](#)
- SCSIID [436](#)
- SCSILUN [437](#)
- VECTOR [439](#)
- DD\_CLASS
  - changing [431](#)
- DD\_MODE
  - changing [426](#)
- DD\_TYPE
  - changing [424](#)
- descriptor name
  - changing mod\_name field [411](#)
- device driver
  - port address offset [421](#)
- DRVR\_NAME
  - changing [430](#)
- FMGR\_NAME
  - changing [429](#)
- LUN
  - changing [422](#)
- MH\_ACCESS
  - changing [412](#)
- MH\_ATTREV
  - changing [417](#)
- MH\_EDITION
  - changing [419](#)
- MH\_GROUP
  - changing [409](#)
- MH\_NAME
  - changing [411](#)
- MH\_TYLAN
  - changing [414](#)
- MH\_USER
  - changing [410](#)
- module header
  - group ID [409](#)
- PD\_SIZE



- changing 423
- PORTADDR
  - changing 421
- user ID
  - module header 410
- sbfdflag
  - changing for SBF 442
- sbfirmqllevel
  - changing for SBF 440
- sbfpriority
  - changing for SBF 441
- sbf\_vector
  - changing for SBF 439
- SCF
  - AUTOECHO
    - changing 388
  - AUTOLF
    - changing 390
  - BAUDRATE
    - changing 226
  - BELLCH
    - changing 380
  - BSB
    - changing 384
  - BSPCH
    - changing 381
  - changing
    - dd\_class field 202
    - dd\_lun field 192
    - dd\_mode field 197
    - dd\_pd\_size field 193
    - dd\_port field 191
    - dd\_type field 194
    - drv\_name field 201
    - hardware\_vector field 207
    - m\_access field 182
    - m\_attr (attributes/revision) field 187

m\_edit field 189  
 m\_group field 179  
 m\_tylan (type/language) field 184  
 m\_user field 180  
 pd\_alf field 390  
 pd\_backsp field 384  
 pd\_bellch field 380  
 pd\_bspch field 381  
 pd\_case field 382  
 pd\_delete field 386  
 pd\_echo field 388  
 pd\_eofch field 378  
 pd\_eorch field 375  
 pd\_inmap0func\_code field 247  
 pd\_inmap0size field 249  
 pd\_inmap0string field 250  
 pd\_inmap0type field 245  
 pd\_inmap10func\_code field 288  
 pd\_inmap10size field 289  
 pd\_inmap10string field 290  
 pd\_inmap10type field 287  
 pd\_inmap11func\_code field 292  
 pd\_inmap11size field 293  
 pd\_inmap11string field 294  
 pd\_inmap11type field 291  
 pd\_inmap12func\_code field 296  
 pd\_inmap12size field 297  
 pd\_inmap12string field 298  
 pd\_inmap12type field 295  
 pd\_inmap13func\_code field 300  
 pd\_inmap13size field 301  
 pd\_inmap13string field 302  
 pd\_inmap13type field 299  
 pd\_inmap14func\_code field 304  
 pd\_inmap14size field 305  
 pd\_inmap14string field 306  
 pd\_inmap14type field 303  
 pd\_inmap15func\_code field 308  
 pd\_inmap15size field 309  
 pd\_inmap15string field 310  
 pd\_inmap15type field 307

pd\_inmap16func\_code field 312  
 pd\_inmap16size field 313  
 pd\_inmap16string field 314  
 pd\_inmap16type field 311  
 pd\_inmap17func\_code field 316  
 pd\_inmap17size field 317  
 pd\_inmap17string field 318  
 pd\_inmap17type field 315  
 pd\_inmap18func\_code field 320  
 pd\_inmap18size field 321  
 pd\_inmap18string field 322  
 pd\_inmap18type field 319  
 pd\_inmap19func\_code field 324  
 pd\_inmap19size field 325  
 pd\_inmap19string field 326  
 pd\_inmap19type field 323  
 pd\_inmap1func\_code field 252  
 pd\_inmap1size field 253  
 pd\_inmap1string field 254  
 pd\_inmap1type field 251  
 pd\_inmap20func\_code field 328  
 pd\_inmap20size field 329  
 pd\_inmap20string field 330  
 pd\_inmap20type field 327  
 pd\_inmap21func\_code field 332  
 pd\_inmap21size field 333  
 pd\_inmap21string field 334  
 pd\_inmap21type field 331  
 pd\_inmap22func\_code field 336  
 pd\_inmap22size field 337  
 pd\_inmap22string field 338  
 pd\_inmap22type field 335  
 pd\_inmap23func\_code field 340  
 pd\_inmap23size field 341  
 pd\_inmap23string field 342  
 pd\_inmap23type field 339  
 pd\_inmap24func\_code field 344  
 pd\_inmap24size field 345  
 pd\_inmap24string field 346  
 pd\_inmap24type field 343  
 pd\_inmap25func\_code field 348

pd\_inmap25size field 349  
 pd\_inmap25string field 350  
 pd\_inmap25type field 347  
 pd\_inmap26func\_code field 352  
 pd\_inmap26size field 353  
 pd\_inmap26string field 354  
 pd\_inmap26type field 351  
 pd\_inmap27func\_code field 356  
 pd\_inmap27size field 357  
 pd\_inmap27string field 358  
 pd\_inmap27type field 355  
 pd\_inmap28func\_code field 360  
 pd\_inmap28size field 361  
 pd\_inmap28string field 362  
 pd\_inmap28type field 359  
 pd\_inmap29func\_code field 364  
 pd\_inmap29size field 365  
 pd\_inmap29string field 366  
 pd\_inmap29type field 363  
 pd\_inmap2func\_code field 256  
 pd\_inmap2size field 257  
 pd\_inmap2string field 258  
 pd\_inmap2type field 255  
 pd\_inmap30func\_code field 368  
 pd\_inmap30size field 369  
 pd\_inmap30string field 370  
 pd\_inmap30type field 367  
 pd\_inmap31func\_code field 372  
 pd\_inmap31size field 373  
 pd\_inmap31string field 374  
 pd\_inmap31type field 371  
 pd\_inmap3func\_code field 260  
 pd\_inmap3size field 261  
 pd\_inmap3string field 262  
 pd\_inmap3type field 259  
 pd\_inmap4func\_code field 264  
 pd\_inmap4size field 265  
 pd\_inmap4string field 266  
 pd\_inmap4type field 263  
 pd\_inmap5func\_code field 268  
 pd\_inmap5size field 269

pd\_inmap5string field 270, 274  
 pd\_inmap5type field 267  
 pd\_inmap6func\_code field 272  
 pd\_inmap6size field 273  
 pd\_inmap6type field 271  
 pd\_inmap7func\_code field 276  
 pd\_inmap7size field 277  
 pd\_inmap7string field 278  
 pd\_inmap7type field 275  
 pd\_inmap8func\_code field 280  
 pd\_inmap8size field 281  
 pd\_inmap8string field 282  
 pd\_inmap8type field 279  
 pd\_inmap9func\_code field 284  
 pd\_inmap9size field 285  
 pd\_inmap9string field 286  
 pd\_inmap9type field 283  
 pd\_insm field 394  
 pd\_nulls field 396  
 pd\_page field 397  
 pd\_pause field 392  
 pd\_tabch field 379  
 pd\_tabsz field 398  
 v\_baud field 226  
 v\_devspec field 235  
 v\_insize field 216  
 v\_intr field 219  
 v\_irqlevel field 208  
 v\_irqmask field 214  
 v\_line field 218  
 v\_lun field 213  
 v\_maxbuff field 215  
 v\_outsize field 217  
 v\_parity field 229  
 v\_pollin field 211  
 v\_pollout field 212  
 v\_priority field 209  
 v\_psch field 223  
 v\_quit field 222  
 v\_rtsstate field 234  
 v\_stopbits field 231

- v\_wordsz field [232](#)
- v\_xoff field [225](#)
- v\_xon field [224](#)
- DD\_CLASS
  - changing [202](#)
- DD\_MODE
  - changing [197](#)
- DD\_TYPE
  - changing [194](#)
- descriptor name
  - changing mod\_name field [181](#)
- device driver
  - port address offset [191](#)
- DRVR\_NAME
  - changing [201](#)
- EOFCH
  - changing [378](#)
- EOLNULLS
  - changing [396](#)
- EORCH
  - changing [375](#)
- FMGR\_NAME
  - changing [200](#)
- FUNC0x01
  - changing [252](#)
- FUNC0x02
  - changing [256](#)
- FUNC0x03
  - changing [260](#)
- FUNC0x04
  - changing [264](#)
- FUNC0x05
  - changing [268](#)
- FUNC0x06
  - changing [272](#)
- FUNC0x07
  - changing [276](#)
- FUNC0x08
  - changing [280](#)
- FUNC0x09
  - changing [284](#)

|          |          |
|----------|----------|
| FUNC0x0a |          |
| changing | 288      |
| FUNC0x0b |          |
| changing | 292      |
| FUNC0x0c |          |
| changing | 296      |
| FUNC0x0d |          |
| changing | 300      |
| FUNC0x0e |          |
| changing | 304      |
| FUNC0x0f |          |
| changing | 308      |
| FUNC0x10 |          |
| changing | 312      |
| FUNC0x11 |          |
| changing | 316      |
| FUNC0x12 |          |
| changing | 320      |
| FUNC0x13 |          |
| changing | 324      |
| FUNC0x14 |          |
| changing | 328      |
| FUNC0x15 |          |
| changing | 332      |
| FUNC0x16 |          |
| changing | 336      |
| FUNC0x17 |          |
| changing | 340      |
| FUNC0x18 |          |
| changing | 344, 348 |
| FUNC0x1a |          |
| changing | 352      |
| FUNC0x1b |          |
| changing | 356      |
| FUNC0x1c |          |
| changing | 360      |
| FUNC0x1d |          |
| changing | 364      |
| FUNC0x1e |          |
| changing | 368      |
| FUNC0x1f |          |

|            |          |
|------------|----------|
| changing   | 372      |
| FUNC0x7f   |          |
| changing   | 247      |
| INPUT_TYPE |          |
| changing   | 211      |
| INSERTMODE |          |
| changing   | 394      |
| INSIZE     |          |
| changing   | 216      |
| IRQ_MASK   |          |
| changing   | 214      |
| IRQLEVEL   |          |
| changing   | 208      |
| KYBDINTR   |          |
| changing   | 219      |
| KYBDPAUSE  |          |
| changing   | 223      |
| KYBDQUIT   |          |
| changing   | 222      |
| LINEDEL    |          |
| changing   | 386      |
| LUN        |          |
| changing   | 192, 213 |
| LUPARITY   |          |
| changing   | 229      |
| MAXBUFF    |          |
| changing   | 215      |
| MH_ACCESS  |          |
| changing   | 182      |
| MH_ATTREV  |          |
| changing   | 187      |
| MH_EDITION |          |
| changing   | 189      |
| MH_GROUP   |          |
| changing   | 179      |
| MH_NAME    |          |
| changing   | 181      |
| MH_TYLAN   |          |
| changing   | 184      |
| MH_USER    |          |
| changing   | 180      |



module header  
     group ID [179](#)  
 OUTPUT\_TYPE  
     changing [212](#)  
 OUTSIZE  
     changing [217](#)  
 PAGE\_SIZE  
     changing [218](#)  
 PAGEPAUSE  
     changing [392](#)  
 PAGESIZE  
     changing [397](#)  
 PD\_SIZE  
     changing [193](#)  
 PORTADDR  
     changing [191](#)  
 PRIORITY  
     changing [209](#)  
 RTSSTATE  
     changing [234](#)  
 SIZE0x01  
     changing [253](#)  
 SIZE0x02  
     changing [257](#)  
 SIZE0x03  
     changing [261](#)  
 SIZE0x04  
     changing [265](#)  
 SIZE0x05  
     changing [269](#)  
 SIZE0x06  
     changing [273](#)  
 SIZE0x07  
     changing [277](#)  
 SIZE0x08  
     changing [281](#)  
 SIZE0x09  
     changing [285](#)  
 SIZE0x0a  
     changing [289](#)  
 SIZE0x0b

|          |     |
|----------|-----|
| changing | 293 |
| SIZE0x0c |     |
| changing | 297 |
| SIZE0x0d |     |
| changing | 301 |
| SIZE0x0e |     |
| changing | 305 |
| SIZE0x0f |     |
| changing | 309 |
| SIZE0x10 |     |
| changing | 313 |
| SIZE0x11 |     |
| changing | 317 |
| SIZE0x12 |     |
| changing | 321 |
| SIZE0x13 |     |
| changing | 325 |
| SIZE0x14 |     |
| changing | 329 |
| SIZE0x15 |     |
| changing | 333 |
| SIZE0x16 |     |
| changing | 337 |
| SIZE0x17 |     |
| changing | 341 |
| SIZE0x18 |     |
| changing | 345 |
| SIZE0x19 |     |
| changing | 349 |
| SIZE0x1a |     |
| changing | 353 |
| SIZE0x1b |     |
| changing | 357 |
| SIZE0x1c |     |
| changing | 361 |
| SIZE0x1d |     |
| changing | 365 |
| SIZE0x1e |     |
| changing | 369 |
| SIZE0x1f |     |
| changing | 373 |

SIZE0x7f  
    changing [249](#)  
STOPBITS  
    changing [231](#)  
STRING0x01  
    changing [254](#)  
STRING0x02  
    changing [258](#)  
STRING0x03  
    changing [262](#)  
STRING0x04  
    changing [266](#)  
STRING0x05  
    changing [270](#)  
STRING0x06  
    changing [274](#)  
STRING0x07  
    changing [278](#)  
STRING0x08  
    changing [282](#)  
STRING0x09  
    changing [286](#)  
STRING0x0a  
    changing [290](#)  
STRING0x0b  
    changing [294](#)  
STRING0x0c  
    changing [298](#)  
STRING0x0d  
    changing [302](#)  
STRING0x0e  
    changing [306](#)  
STRING0x0f  
    changing [310](#)  
STRING0x10  
    changing [314](#)  
STRING0x11  
    changing [318](#)  
STRING0x12  
    changing [322](#)  
STRING0x13

|            |     |
|------------|-----|
| changing   | 326 |
| STRING0x14 |     |
| changing   | 330 |
| STRING0x15 |     |
| changing   | 334 |
| STRING0x16 |     |
| changing   | 338 |
| STRING0x17 |     |
| changing   | 342 |
| STRING0x18 |     |
| changing   | 346 |
| STRING0x19 |     |
| changing   | 350 |
| STRING0x1a |     |
| changing   | 354 |
| STRING0x1b |     |
| changing   | 358 |
| STRING0x1c |     |
| changing   | 362 |
| STRING0x1d |     |
| changing   | 366 |
| STRING0x1e |     |
| changing   | 370 |
| STRING0x1f |     |
| changing   | 374 |
| STRING0x7f |     |
| changing   | 250 |
| TABCH      |     |
| changing   | 379 |
| TABSIZE    |     |
| changing   | 398 |
| TYPE0x01   |     |
| changing   | 251 |
| TYPE0x02   |     |
| changing   | 255 |
| TYPE0x03   |     |
| changing   | 259 |
| TYPE0x04   |     |
| changing   | 263 |
| TYPE0x05   |     |
| changing   | 267 |

|          |                     |
|----------|---------------------|
| TYPE0x06 |                     |
| changing | <a href="#">271</a> |
| TYPE0x07 |                     |
| changing | <a href="#">275</a> |
| TYPE0x08 |                     |
| changing | <a href="#">279</a> |
| TYPE0x09 |                     |
| changing | <a href="#">283</a> |
| TYPE0x0a |                     |
| changing | <a href="#">287</a> |
| TYPE0x0b |                     |
| changing | <a href="#">291</a> |
| TYPE0x0c |                     |
| changing | <a href="#">295</a> |
| TYPE0x0d |                     |
| changing | <a href="#">299</a> |
| TYPE0x0e |                     |
| changing | <a href="#">303</a> |
| TYPE0x0f |                     |
| changing | <a href="#">307</a> |
| TYPE0x10 |                     |
| changing | <a href="#">311</a> |
| TYPE0x11 |                     |
| changing | <a href="#">315</a> |
| TYPE0x12 |                     |
| changing | <a href="#">319</a> |
| TYPE0x13 |                     |
| changing | <a href="#">323</a> |
| TYPE0x14 |                     |
| changing | <a href="#">327</a> |
| TYPE0x15 |                     |
| changing | <a href="#">331</a> |
| TYPE0x16 |                     |
| changing | <a href="#">335</a> |
| TYPE0x17 |                     |
| changing | <a href="#">339</a> |
| TYPE0x18 |                     |
| changing | <a href="#">343</a> |
| TYPE0x19 |                     |
| changing | <a href="#">347</a> |
| TYPE0x1a |                     |

- changing 351
- TYPE0x1b
  - changing 355
- TYPE0x1c
  - changing 359
- TYPE0x1d
  - changing 363
- TYPE0x1e
  - changing 367
- TYPE0x1f
  - changing 371
- TYPE0x7f
  - changing 245
- UPC\_LICK
  - changing 382
- user ID
  - module header 180
- VECTOR
  - changing 207
- WORDSIZE
  - changing 232
- XOFF
  - changing 225
- XON
  - changing 224
- SCF baud rate,
  - changing the 226, 229
- SCF request to send flag,
  - changing the 234
- SCF stop bits,
  - changing the 231
- SCSIID
  - changing
    - for SBF 436
- SCSILUN
  - changing 437
    - for PCF 572
    - for RBF 506
- segment
  - allocation size of
    - changing 485, 551

## SEGSIZE

- changing [485](#), [551](#)

## setting module permissions

- cnfgdata [21](#)

- init [113](#)

- PCF [522](#)

- pipe [588](#)

- RBF [456](#)

- SBF [412](#)

- SCF [182](#)

- shell [108](#)

## SIDES

- changing [478](#), [544](#)

## sides

- number of disk

- changing [478](#), [544](#)

## SITE

- changing

- for init [124](#)

## size of

- path descriptor

- changing [193](#), [423](#), [467](#), [533](#), [599](#)

## SIZE0x01

- changing

- for SCF [253](#)

## SIZE0x02

- changing

- for SCF [257](#)

## SIZE0x03

- changing

- for SCF [261](#)

## SIZE0x04

- changing

- for SCF [265](#)

## SIZE0x05

- changing

- for SCF [269](#)

## SIZE0x06

- changing

- for SCF [273](#)

## SIZE0x07

|                     |     |
|---------------------|-----|
| changing<br>for SCF | 277 |
| SIZE0x08            |     |
| changing<br>for SCF | 281 |
| SIZE0x09            |     |
| changing<br>for SCF | 285 |
| SIZE0x0a            |     |
| changing<br>for SCF | 289 |
| SIZE0x0b            |     |
| changing<br>for SCF | 293 |
| SIZE0x0c            |     |
| changing<br>for SCF | 297 |
| SIZE0x0d            |     |
| changing<br>for SCF | 301 |
| SIZE0x0e            |     |
| changing<br>for SCF | 305 |
| SIZE0x0f            |     |
| changing<br>for SCF | 309 |
| SIZE0x10            |     |
| changing<br>for SCF | 313 |
| SIZE0x11            |     |
| changing<br>for SCF | 317 |
| SIZE0x12            |     |
| changing<br>for SCF | 321 |
| SIZE0x13            |     |
| changing<br>for SCF | 325 |
| SIZE0x14            |     |
| changing<br>for SCF | 329 |



SIZE0x15  
    changing  
        for SCF [333](#)

SIZE0x16  
    changing  
        for SCF [337](#)

SIZE0x17  
    changing  
        for SCF [341](#)

SIZE0x18  
    changing  
        for SCF [345](#)

SIZE0x19  
    changing  
        for SCF [349](#)

SIZE0x1a  
    changing  
        for SCF [353](#)

SIZE0x1b  
    changing  
        for SCF [357](#)

SIZE0x1c  
    changing  
        for SCF [361](#)

SIZE0x1d  
    changing  
        for SCF [365](#)

SIZE0x1e  
    changing  
        for SCF [369](#)

SIZE0x1f  
    changing  
        for SCF [373](#)

SIZE0x7f  
    changing  
        for SCF [249](#)

SLICE  
    changing  
        for init [141](#)

software interrupt  
    changing [441](#), [501](#), [567](#)

- sparam\_string
  - changing
    - for init 129
- startup 107
- STEP
  - changing
    - for PCF 569
    - for RBF 503
- step rate
  - of drive heads
    - changing 503, 569
- sticky
  - module
    - setting value for cnfgdata 27
    - setting value for init 119
    - setting value for PCF 528
    - setting value for pipe 594
    - setting value for RBF 462
    - setting value for SBF 418
    - setting value for SCF 188
- stop bits
  - changing for SCF device 231
- STOPBITS
  - changing
    - for SCF 231
- STRING0x01
  - changing
    - for SCF 254
- STRING0x02
  - changing
    - for SCF 258
- STRING0x03
  - changing
    - for SCF 262
- STRING0x04
  - changing
    - for SCF 266
- STRING0x05
  - changing
    - for SCF 270
- STRING0x06

|                     |     |
|---------------------|-----|
| changing<br>for SCF | 274 |
| STRING0x07          |     |
| changing<br>for SCF | 278 |
| STRING0x08          |     |
| changing<br>for SCF | 282 |
| STRING0x09          |     |
| changing<br>for SCF | 286 |
| STRING0x0a          |     |
| changing<br>for SCF | 290 |
| STRING0x0b          |     |
| changing<br>for SCF | 294 |
| STRING0x0c          |     |
| changing<br>for SCF | 298 |
| STRING0x0d          |     |
| changing<br>for SCF | 302 |
| STRING0x0e          |     |
| changing<br>for SCF | 306 |
| STRING0x0f          |     |
| changing<br>for SCF | 310 |
| STRING0x10          |     |
| changing<br>for SCF | 314 |
| STRING0x11          |     |
| changing<br>for SCF | 318 |
| STRING0x12          |     |
| changing<br>for SCF | 322 |
| STRING0x13          |     |
| changing<br>for SCF | 326 |

STRING0x14  
    changing  
        for SCF 330

STRING0x15  
    changing  
        for SCF 334

STRING0x16  
    changing  
        for SCF 338

STRING0x17  
    changing  
        for SCF 342

STRING0x18  
    changing  
        for SCF 346

STRING0x19  
    changing  
        for SCF 350

STRING0x1a  
    changing  
        for SCF 354

STRING0x1b  
    changing  
        for SCF 358

STRING0x1c  
    changing  
        for SCF 362

STRING0x1d  
    changing  
        for SCF 366

STRING0x1e  
    changing  
        for SCF 370

STRING0x1f  
    changing  
        for SCF 374

STRING0x7f  
    changing  
        for SCF 250

subnet\_mask  
    changing

- for cnfgdata 74
- interface data
  - configuration 74
- subroutine
  - module header
    - type code for cnfgdata 24
    - type code for init 116
    - type code for PCF 525
    - type code for pipe 591
    - type code for RBF 459
    - type code for SBF 415
    - type code for SCF 185
- super user only
  - module
    - setting value for cnfgdata 27
    - setting value for init 119
    - setting value for PCF 528
    - setting value for pipe 594
    - setting value for RBF 462
    - setting value for SBF 418
    - setting value for SCF 188
- SYS\_DEVICE
  - changing
    - for init 130
- SYS\_PARAMS
  - changing
    - for init 129
- SYS\_PRIOR
  - changing
    - for init 142
- SYS\_START
  - changing
    - for init 128
- SYS\_TMZONE
  - changing
    - for init 147
- sysboot file 107
- sysgo 108
- sysgo\_name
  - changing
    - for init 128

- system
  - time zone 147
- system module
  - module header
    - type code for cnfgdata 24
    - type code for init 116
    - type code for PCF 525
    - type code for pipe 591
    - type code for RBF 459
    - type code for SBF 415
    - type code for SCF 185
- system-state
  - module
    - setting value for cnfgdata 27
    - setting value for init 119
    - setting value for PCF 528
    - setting value for pipe 594
    - setting value for RBF 462
    - setting value for SBF 418
    - setting value for SCF 188

---

T

- tab character
  - changing 379
- TABCH
  - changing
    - for SCF 379
- TABSIZE
  - changing
    - for SCF 398
- Tape Device Logical Unit Number
  - changing 437
- TICK\_NAME
  - changing
    - for init 133
- TICK\_SEC
  - changing
    - for init 140
- ticker\_name
  - changing

- for init 133
- total number of cylinders
  - setting 508
- TOTCYLS
  - changing
    - for RBF 508
- track
  - number of blocks per
    - changing 483, 549
- track 0
  - number of blocks per
    - changing 484, 550
- track offset
  - changing 487, 553
- transfer memory size
  - changing 497
- trap library
  - module header
    - type code for cnfgdata 24
    - type code for init 116
    - type code for PCF 525
    - type code for pipe 591
    - type code for RBF 459
    - type code for SBF 415
    - type code for SCF 185
- TRKOFFS
  - changing
    - for PCF 553
    - for RBF 487
- TRYS
  - changing 489, 555
- type
  - changing
    - for init 157
  - of module
    - setting for cnfgdata 23
    - setting for init 115
    - setting for PCF 524
    - setting for pipe 590
    - setting for RBF 458
    - setting for SBF 414

|                 |     |
|-----------------|-----|
| setting for SCF | 184 |
| TYPE0x01        |     |
| changing        |     |
| for SCF         | 251 |
| TYPE0x02        |     |
| changing        |     |
| for SCF         | 255 |
| TYPE0x03        |     |
| changing        |     |
| for SCF         | 259 |
| TYPE0x04        |     |
| changing        |     |
| for SCF         | 263 |
| TYPE0x05        |     |
| changing        |     |
| for SCF         | 267 |
| TYPE0x06        |     |
| changing        |     |
| for SCF         | 271 |
| TYPE0x07        |     |
| changing        |     |
| for SCF         | 275 |
| TYPE0x08        |     |
| changing        |     |
| for SCF         | 279 |
| TYPE0x09        |     |
| changing        |     |
| for SCF         | 283 |
| TYPE0x0a        |     |
| changing        |     |
| for SCF         | 287 |
| TYPE0x0b        |     |
| changing        |     |
| for SCF         | 291 |
| TYPE0x0c        |     |
| changing        |     |
| for SCF         | 295 |
| TYPE0x0d        |     |
| changing        |     |
| for SCF         | 299 |
| TYPE0x0e        |     |



|                                 |     |
|---------------------------------|-----|
| changing<br>for SCF             | 303 |
| TYPE0x0f<br>changing<br>for SCF | 307 |
| TYPE0x10<br>changing<br>for SCF | 311 |
| TYPE0x11<br>changing<br>for SCF | 315 |
| TYPE0x12<br>changing<br>for SCF | 319 |
| TYPE0x13<br>changing<br>for SCF | 323 |
| TYPE0x14<br>changing<br>for SCF | 327 |
| TYPE0x15<br>changing<br>for SCF | 331 |
| TYPE0x16<br>changing<br>for SCF | 335 |
| TYPE0x17<br>changing<br>for SCF | 339 |
| TYPE0x18<br>changing<br>for SCF | 343 |
| TYPE0x19<br>changing<br>for SCF | 347 |
| TYPE0x1a<br>changing<br>for SCF | 351 |
| TYPE0x1b<br>changing<br>for SCF | 355 |

TYPE0x1c  
    changing  
        for SCF   359

TYPE0x1d  
    changing  
        for SCF   363

TYPE0x1e  
    changing  
        for SCF   367

TYPE0x1f  
    changing  
        for SCF   371

TYPE0x7f  
    changing  
        for SCF   245

---

U

UPC\_LOCK  
    changing  
        for SCF   382

upper-case lock character  
    changing   382

USRACCT\_NAME  
    changing  
        for init   136

utility  
    chd   130  
    chx   130

---

V

v\_baud  
    changing  
        for SCF   226

v\_devspec  
    changing  
        for SCF   235

v\_insize  
    changing

for SCF [216](#)  
 v\_intr  
     changing  
         for SCF [219](#)  
 v\_irqlevel  
     changing  
         for PCF [566](#)  
         for RBF [500](#)  
         for SCF [208](#)  
 v\_irqmask  
     changing  
         for SCF [214](#)  
 v\_line  
     changing  
         for SCF [218](#)  
 v\_lun  
     changing  
         for SCF [213](#)  
 v\_maxbuff  
     changing  
         for SCF [215](#)  
 v\_outsize  
     changing  
         for SCF [217](#)  
 v\_parity  
     changing  
         for SCF [229](#)  
 v\_pollin  
     changing  
         for SCF [211](#)  
 v\_pollout field  
     changing  
         for SCF [212](#)  
 v\_priority  
     changing  
         for PCF [567](#)  
         for RBF [501](#)  
         for SCF [209](#)  
 v\_psch  
     changing  
         for SCF [223](#)

- v\_quit
  - changing
    - for SCF [222](#)
- v\_rtsstate
  - changing
    - for SCF [234](#)
- v\_stopbits
  - changing
    - for SCF [231](#)
- v\_vector
  - changing
    - for PCF [565](#)
    - for RBF [499](#)
- v\_wordsize
  - changing
    - for SCF [232](#)
- v\_xoff
  - changing
    - for SCF [225](#)
- v\_xon
  - changing
    - for SCF [224](#)
- VECTOR
  - changing
    - for PCF [565](#)
    - for RBF [499](#)
    - for SBF [439](#)
    - for SCF [207](#)
- vector interrupt
  - changing [439](#), [499](#), [565](#)
- vector number
  - setting
    - for console device [31](#), [48](#)
- VERIFY
  - changing
    - for PCF [545](#)
    - for RBF [479](#)
- verify
  - write operation
    - changing [479](#), [545](#)

---

W

## wildcard

## module header

- language code for cnfgdata 25
- language code for init 117
- language code for PCF 526
- language code for pipe 592
- language code for RBF 460
- language code for SBF 416
- language code for SCF 186
- type code for cnfgdata 23
- type code for init 116
- type code for PCF 525
- type code for pipe 591
- type code for RBF 459
- type code for SBF 415
- type code for SCF 185

## WORDSIZE

## changing

- for SCF 232

## wordsize

- changing for SCF device 232

## write precompensation

## starting point

- changing 493, 559

---

X

## XFERSIZE

- changing 497, 563

## XOFF

## changing

- for SCF 225

## X-OFF control key

- changing for keyboard 225

## XON

## changing

- for SCF 224

## X-ON control key

- changing for keyboard 224



---

# Product Discrepancy Report

---

To: Microware Customer Support

FAX: 515-224-1352

From: \_\_\_\_\_

Company: \_\_\_\_\_

Phone: \_\_\_\_\_

Fax: \_\_\_\_\_ Email: \_\_\_\_\_

Product Name: OS-9

Description of Problem:

---

---

---

---

---

---

---

---

---

---

---

---

Host Platform \_\_\_\_\_

Target Platform \_\_\_\_\_



MICROWARE SOFTWARE