

# New Chapters

---

Edition 4.10.1, for RTEMS 4.10.1

18 November 2011

On-Line Applications Research Corporation

---

COPYRIGHT © 1988 - 2011.  
On-Line Applications Research Corporation (OAR).

The authors have used their best efforts in preparing this material. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. No warranty of any kind, expressed or implied, with regard to the software or the material contained in this document is provided. No liability arising out of the application or use of any product described in this document is assumed. The authors reserve the right to revise this material and to make changes from time to time in the content hereof without obligation to notify anyone of such revision or changes.

The RTEMS Project is hosted at <http://www.rtems.com>. Any inquiries concerning RTEMS, its related support components, its documentation, or any custom services for RTEMS should be directed to the contacts listed on that site. A current list of RTEMS Support Providers is at <http://www.rtems.com/support.html>.

# Table of Contents

<b>1</b>	<b>Error Reporting Support .....</b>	<b>1</b>
1.1	Introduction .....	1
1.2	Background .....	1
1.2.1	Error Handling in an Embedded System .....	1
1.3	Operations .....	1
1.3.1	Reporting an Error .....	1
1.4	Routines .....	1
1.4.1	rtems_status_text - ASCII Version of RTEMS Status .....	2
1.4.2	rtems_error - Report an Error .....	3
1.4.3	rtems_panic - Report an Error and Panic .....	4
<b>2</b>	<b>Monitor Task .....</b>	<b>5</b>
2.1	Introduction .....	5
2.2	Background .....	5
2.3	Operations .....	5
2.3.1	Initializing the Monitor .....	5
2.4	Routines .....	5
2.4.1	rtems_monitor_init - Initialize the Monitor Task .....	6
2.4.2	rtems_monitor_wakeup - Wakeup the Monitor Task .....	7
2.5	Monitor Interactive Commands .....	8
2.5.1	help - Obtain Help .....	8
2.5.2	pause - Pause Monitor for a Specified Number of Ticks .....	8
2.5.3	exit - Invoke a Fatal RTEMS Error .....	8
2.5.4	symbol - Show Entries from Symbol Table .....	8
2.5.5	continue - Put Monitor to Sleep Waiting for Explicit Wakeup .....	8
2.5.6	config - Show System Configuration .....	8
2.5.7	itask - List Init Tasks .....	9
2.5.8	mpci - List MPCIE Config .....	9
2.5.9	task - Show Task Information .....	9
2.5.10	queue - Show Message Queue Information .....	9
2.5.11	extension - User Extensions .....	9
2.5.12	driver - Show Information About Named Drivers .....	9
2.5.13	dname - Show Information About Named Drivers .....	9
2.5.14	object - Generic Object Information .....	9
2.5.15	node - Specify Default Node for Commands That Take IDs .....	9
	<b>Command and Variable Index .....</b>	<b>11</b>
	<b>Concept Index .....</b>	<b>13</b>



# 1 Error Reporting Support

## 1.1 Introduction

These error reporting facilities are an RTEMS support component that provide convenient facilities for handling error conditions in an RTEMS application. of each task using a period. The services provided by the error reporting support component are:

- `rtems_error` - Report an Error
- `rtems_panic` - Report an Error and Panic
- `rtems_status_text` - ASCII Version of RTEMS Status

## 1.2 Background

### 1.2.1 Error Handling in an Embedded System

Error handling in an embedded system is a difficult problem. If the error is severe, then the only recourse is to shut the system down in a safe manner. Other errors can be detected and compensated for. The error reporting routines in this support component – `rtems_error` and `rtems_panic` assume that if the error is severe enough, then the system should be shutdown. If a simple shutdown with some basic diagnostic information is not sufficient, then these routines should not be used in that particular system. In this case, use the `rtems_status_text` routine to construct an application specific error reporting routine.

## 1.3 Operations

### 1.3.1 Reporting an Error

The `rtems_error` and `rtems_panic` routines can be used to print some diagnostic information and shut the system down. The `rtems_error` routine is invoked with a user specified error level indicator. This error indicator is used to determine if the system should be shutdown after reporting this error.

## 1.4 Routines

This section details the error reporting support component's routine. A subsection is dedicated to each of this manager's routines and describes the calling sequence, related constants, usage, and status codes.

### 1.4.1 rtems\_status\_text - ASCII Version of RTEMS Status

#### CALLING SEQUENCE:

```
const char *rtems_status_text(  
    rtems_status_code status  
);
```

#### STATUS CODES:

Returns a pointer to a constant string that describes the given RTEMS status code.

#### DESCRIPTION:

This routine returns a pointer to a string that describes the RTEMS status code specified by `status`.

#### NOTES:

NONE

### 1.4.2 rtems\_error - Report an Error

#### CALLING SEQUENCE:

```
int rtems_error(  
    int          error_code,  
    const char *printf_format,  
    ...  
);
```

#### STATUS CODES:

Returns the number of characters written.

#### DESCRIPTION:

This routine prints the requested information as specified by the `printf_format` parameter and the zero or more optional arguments following that parameter. The `error_code` parameter is an error number with either `RTEMS_ERROR_PANIC` or `RTEMS_ERROR_ABORT` bitwise or'ed with it. If the `RTEMS_ERROR_PANIC` bit is set, then then the system is system is shutdown via a call to `_exit`. If the `RTEMS_ERROR_ABORT` bit is set, then then the system is system is shutdown via a call to `abort`.

#### NOTES:

NONE

### 1.4.3 rtems\_panic - Report an Error and Panic

#### CALLING SEQUENCE:

```
int rtems_panic(  
    const char *printf_format,  
    ...  
);
```

#### STATUS CODES:

Returns the number of characters written.

#### DESCRIPTION:

This routine is a wrapper for the `rtems_error` routine with an implied error level of `RTEMS_ERROR_PANIC`. See `rtems_error` for more information.

#### NOTES:

NONE

## 2 Monitor Task

### 2.1 Introduction

The monitor task is a simple interactive shell that allows the user to make inquiries about the state of various system objects. The routines provided by the monitor task manager are:

- `rtems_monitor_init` - Initialize the Monitor Task
- `rtems_monitor_wakeup` - Wakeup the Monitor Task

### 2.2 Background

There is no background information.

### 2.3 Operations

#### 2.3.1 Initializing the Monitor

The monitor is initialized by calling `rtems_monitor_init`. When initialized, the monitor is created as an independent task. An example of initializing the monitor is shown below:

```
#include <rtems/monitor.h>

...

rtems_monitor_init(0);
```

The "0" parameter to the `rtems_monitor_init` routine causes the monitor to immediately enter command mode. This parameter is a bitfield. If the monitor is to suspend itself on startup, then the `RTEMS_MONITOR_SUSPEND` bit should be set.

### 2.4 Routines

This section details the monitor task manager's routines. A subsection is dedicated to each of this manager's routines and describes the calling sequence, related constants, usage, and status codes.

### 2.4.1 rtems\_monitor\_init - Initialize the Monitor Task

#### CALLING SEQUENCE:

```
void rtems_monitor_init(  
    unsigned32 monitor_flags  
);
```

#### STATUS CODES: NONE

#### DESCRIPTION:

This routine initializes the RTEMS monitor task. The `monitor_flags` parameter indicates how the server task is to start. This parameter is a bitfield and has the following constants associated with it:

- **RTEMS\_MONITOR\_SUSPEND** - suspend monitor on startup
- **RTEMS\_MONITOR\_GLOBAL** - monitor should be global

If the `RTEMS_MONITOR_SUSPEND` bit is set, then the monitor task will suspend itself after it is initialized. A subsequent call to `rtems_monitor_wakeup` will be required to activate it.

#### NOTES:

The monitor task is created with priority 1. If there are application tasks at priority 1, then there may be times when the monitor task is not executing.

### 2.4.2 rtems\_monitor\_wakeup - Wakeup the Monitor Task

#### CALLING SEQUENCE:

```
void rtems_monitor_wakeup( void );
```

#### STATUS CODES: NONE

#### DESCRIPTION:

This routine is used to activate the monitor task if it is suspended.

#### NOTES:

NONE

## 2.5 Monitor Interactive Commands

The following commands are supported by the monitor task:

- **help** - Obtain Help
- **pause** - Pause Monitor for a Specified Number of Ticks
- **exit** - Invoke a Fatal RTEMS Error
- **symbol** - Show Entries from Symbol Table
- **continue** - Put Monitor to Sleep Waiting for Explicit Wakeup
- **config** - Show System Configuration
- **itask** - List Init Tasks
- **mpci** - List MPCI Config
- **task** - Show Task Information
- **queue** - Show Message Queue Information
- **extension** - User Extensions
- **driver** - Show Information About Named Drivers
- **dname** - Show Information About Named Drivers
- **object** - Generic Object Information
- **node** - Specify Default Node for Commands That Take IDs

### 2.5.1 help - Obtain Help

The **help** command prints out the list of commands. If invoked with a command name as the first argument, detailed help information on that command is printed.

### 2.5.2 pause - Pause Monitor for a Specified Number of Ticks

The **pause** command cause the monitor task to suspend itself for the specified number of ticks. If this command is invoked with no arguments, then the task is suspended for 1 clock tick.

### 2.5.3 exit - Invoke a Fatal RTEMS Error

The **exit** command invokes `rtems_error_occurred` directive with the specified error code. If this command is invoked with no arguments, then the `rtems_error_occurred` directive is invoked with an arbitrary error code.

### 2.5.4 symbol - Show Entries from Symbol Table

The **symbol** command lists the specified entries in the symbol table. If this command is invoked with no arguments, then all the symbols in the symbol table are printed.

### 2.5.5 continue - Put Monitor to Sleep Waiting for Explicit Wakeup

The **continue** command suspends the monitor task with no timeout.

### 2.5.6 config - Show System Configuration

The **config** command prints the system configuration.

### **2.5.7 itask - List Init Tasks**

The `itask` command lists the tasks in the initialization tasks table.

### **2.5.8 mpci - List MPCl Config**

The `mpci` command shows the MPCl configuration information

### **2.5.9 task - Show Task Information**

The `task` command prints out information about one or more tasks in the system. If invoked with no arguments, then information on all the tasks in the system is printed.

### **2.5.10 queue - Show Message Queue Information**

The `queue` command prints out information about one or more message queues in the system. If invoked with no arguments, then information on all the message queues in the system is printed.

### **2.5.11 extension - User Extensions**

The `extension` command prints out information about the user extensions.

### **2.5.12 driver - Show Information About Named Drivers**

The `driver` command prints information about the device driver table.

### **2.5.13 dname - Show Information About Named Drivers**

The `dname` command prints information about the named device drivers.

### **2.5.14 object - Generic Object Information**

The `object` command prints information about RTEMS objects.

### **2.5.15 node - Specify Default Node for Commands That Take IDs**

The `node` command sets the default node for commands that look at object ID ranges.



## Command and Variable Index

There are currently no Command and Variable Index entries.



## Concept Index

There are currently no Concept Index entries.

