Rtfs

Configuration Guide

Revised October 2008

For best online viewing experience we recommend using Adobe Acrobat's **Bookmarks** tab for navigating

EBS Inc. 39 Court Street Groton MA 01450 USA

http://www.ebsembeddedsoftware.com

Table of Contents

Synopsis

- This document describes Rtfs configuration values that may be modified.
- Compile time configuration – Conditional compilation is used to customize Rtfs for the target environment and to select features to include in the build. The following files contain compile time configuration values:
    - Compiler and CPU configurations - rtfscommon/include/rtfsarch.h
    - Feature set configurations - rtfscommon/include/rtfsconf.h
    - Device driver selection - rtfscommon/include/rtfsconf.h

- Run time configuration – Rtfs buffering configuration, operating policy selection and device driver attachment is done at run-time. See the Initialization and shutdown and Media driver interface sections of the API reference manual for more information on run time configuration options.

Compile time compiler and architecture configuration

These architecture specific configuration constants are provided in: rtfscommon/include/rtfsarch.h *You must check and, if necessary, modify these definitions for your architecture*	
Constant	Setting
KS_LITTLE_ENDIAN	Set this value to 1 if your target device has little endian byte order. An example of a little endian target is the Intel Pentium, and example of a non little endian target is Motorola ColdFire.
KS_LITTLE_ODD_PTR_OK	Set this value to 1 if your architecture is little endian and it can dereference word, and dword pointers on any address boundary. An example of a little endian target that can dereference these pointers on any boundary is the Intel Pentium. An example of a little endian target that can not dereference these pointers on any boundary is the MIPS processor.
RTFS_WINDOWS	Enable this if using Microsoft Windows. This constant and RTFS_LINUX are used sparingly to configure the emulation host disk and raw disk drivers and the

	telnet server module. All uses are all optional and can be disabled or worked around in other systems.
RTFS_LINUX	Enable this if using Linux.
INCLUDE_DEBUG_TRUE_ASSERT	Asserts for unexpected conditions are compiled into Rtfs using the macros **ERTFS_ASSERT(X)** and **RTFS_ASSERT_TEST(X)** *see rtfsarch.h*. if **INCLUDE_DEBUG_TRUE_ASSERT** is enabled then these asserts use the compiler's assert((X)); call otherwise they result in callbacks to **rtfs_diag_callback()** with arguments **RTFS_CBD_ASSERT** and **RTFS_CBD_ASSERT_TEST** respectively.
INCLUDE_THREAD_SETENV_SUPPORT	Set this to 1 if thread local storage is supported. Thread thread local storage provides an efficient way for Rtfs to bind user context structures to the threads using Rtfs. *Note: If this option is enabled two porting layer functions must be provided see the porting guide for more information on* ***rtfs_port_set_task_env()*** *and* ***rtfs_port_get_task_env()****.*
INCLUDE_THREAD_EXIT_CALLBACK	Set this to 1 if Rtfs can make a callback when a task exits or is destroyed. *Note: If this option is a porting layer function must be provided. See the porting guide for more information on* ***rtfs_port_set_task_exit_handler()****.* If this function is not available Rtfs the application must call **pc_free_user()** before a threads exit or Rtfs will run out of user structures.
INCLUDE_NATIVE_64_TYPE	Enable this if your compiler supports bit integers. If this value is one, the M64XXX() macro package is implemented using native operators

	otherwise the macro package operates on the ddword 64 bit integer meta-structure.
ddword	If **INCLUDE_NATIVE_64_TYPE** is set to 1 you must set this to your compiler's native 64 bit integer type. If **INCLUDE_NATIVE_64_TYPE** is set to 0 a ddword typedef is provided that consists of two dwords. The default definition is: **#define ddword unsigned long long**

The following table contains additional element from rtfsarch.h that should rarely, if ever need to be changed.	
Byte (8 bit unsigned)	typedef unsigned char byte;
Word (16 bit unsigned)	typedef unsigned short word;
Dword (32 bit unsigned)	typedef unsigned long dword;
Boolean (TRUE, FALSE)	#define BOOLEAN int
TRUE	#define TRUE 1
FALSE	#define FALSE 0
KS_CONSTANT (const declaration)	#define KS_CONSTANT const

Compile time feature set configuration

These compile time options are defined in: rtfscommon/include/rtfsconf.h Modify values in this file to enable and disable features or Rtfs	
Constant	Setting
The following configuration constants are available for all configurations of Rtfs	
INCLUDE_CS_JIS	Set to 1 to support Japanese Language
SUPPORT_EXTENDED_PARTITIONS	If 1 Rtfs will include code to interpret disks with extended partitions and to create extended DOS partitions if more

	than 4 partitions on a single device is required.
RTFS_CFG_MAX_DIRENTS	Set to the maximum number of directory entries allowed per subdirectory. The default value, 32768, is very large, but sufficient to force breaking out of endless loops. Reduce the number if a more conservative maximum is desired. The ***RTFS_CFG_MAX_DIRENTS*** policy is enforced at block boundaries so slightly more than the dictated maximum may be created.
The following configuration constants are included in Rtfsconf.h but are meaningful only when the Failsafe Journaling option has been purchased.	
INCLUDE_FAILSAFE_CODE	Include Failsafe
INCLUDE_TRANSACTION_FILES	Include transaction file support. Also requires **INCLUDE_FAILSAFE_CODE** (see **pc_efilio_open()**).
The following configuration constants are included in Rtfsconf.h but are not meaningful if only the RtfsBasic configuration has been purchased.	
INCLUDE_CS_UNICODE	Set to 1 to support Unicode characters. *Note: If Unicode is enabled many API calls have a counterpart API that processes Unicode arguments and returns strings in Unicode. These APIs have the suffix **_uc** and are documented along with the API reference guide.*
INCLUDE_VFAT	Include long file name support
INCLUDE_FAT16	Include FAT12 and FAT16 support
INCLUDE_FAT32	Include FAT32 support
INCLUDE_FAT32	Include EXFAT support
INCLUDE_RTFS_FREEMANAGER	Enable to include a memory based free manager. This feature eliminates the need to scan the FAT table to allocate clusters. When this feature is enabled all cluster allocations occur in "real time", This greatly improves operating

	speed and makes extending data files deterministic, eliminating the stalls that can otherwise occur when extending data files.
The following configuration constants are included in Rtfsconf.h but are meaningful only for RtfsProPlus configurations. They are not used for RtfsBasic or RtfsPro.	
INCLUDE_ASYNCRONOUS_API	Enable to include the asynchronous API calls described in the API reference guide.
INCLUDE_DEBUG_TEST_CODE	Enable this to include additional compile time code required to perform package regression tests. The basic regression test does not require this to be set.
INCLUDE_DEBUG_RUNTIME_STATS	If **INCLUDE_DEBUG_RUNTIME_STATS** additional statistics are accumulated while Rtfs is running that may be accessed be calling **pc_diskio_runtime_stats()**. These are useful during application development to determine if your disk access patterns are optimal. See manual page for be calling **pc_diskio_runtime_stats()** for a description of what statistics are available. *Note: Enabling this option does not consume a lot of additional resources, requiring a few hundred bytes of additional ram per drive and very negligible additional code space and run time overhead.* If you wish to use the monitoring features of **pc_diskio_runtime_stats()** in your product you may consider leaving **INCLUDE_DEBUG_RUNTIME_STATS** enabled.

Run time memory configuration

Run time configuration – Rtfs buffering configuration, operating policy selection and device driver attachment is done at run-time. See the Initialization and shutdown and Media driver interface sections of the API reference manual for more information on run time configuration options.

Compile time device driver selection

Device drivers provided with Rtfs may be enabled by modifying several constants at the end of the file rtfscommon/include/rtfsconf.h.

Notes:

- *Externally provided device drivers must be attached using procedures outlined in the Driver and Porting Guide. These following compile time setting are required to enable and disable Rtfs supplied device drivers only Rtfs uses the compile time constants to conditionally include or exclude the device driver from the standard build and to enable certain features in target specific porting layer files, the main source code does not reference these constants.*
- *Some target specific modifications to the porting layer will be needed when certain devices are enabled. See the RTFS porting guide for more information.*

	Result if set to 1	Notes
INCLUDE_IDE	Include IDE driver	Requires modifications to portkern.c and portio.c
INCLUDE_PCMCIA	Include PCMCIA driver	Requires modifications to portkern.c and portio.c. Also requires **INCLUDE_82365_PCMCTRL** or an alternate controller implementation
INCLUDE_PCMCIA_SRAM	Include PCMCIA static ram card driver	Requires **INCLUDE_PCMCIA.**
INCLUDE_COMPACT_FLASH	Support compact flash	Requires **INCLUDE_IDE**, requires **INCLUDE_PCMCIA** if not using **TRUE-IDE** mode.
INCLUDE_FLASH_FTL	Include linear flash driver	Includes support for several Intel flash parts as well as ram and disk based emulation. See drflsmtd.c**.**
INCLUDE_ROMDISK	Include rom	Rom Disk images may be

	disk driver	generated from Windows subdirectories with the mkrom tool.
INCLUDE_RAMDISK	Include RAM disk driver	The constants **NUM_RAMDISK_PAGES** and **RAMDISK_PAGE_SIZE** in drramdisk.c determine the size of the ram disk.
INCLUDE_SMARTMEDIA	Include smart Media driver	Requires modifications to portkern.c and portio.c.
INCLUDE_FLOPPY	Include floppy disk driver	Requires modifications to portkern.c and portio.c. Supports only PC architectures, other architectures require customization.
INCLUDE_HOSTDISK	Include host disk simulator	Available only for Windows and Linux desktop emulation platforms.
INCLUDE_HOSTDEV	Include raw access to disks under windows	Available only for Windows and Linux platforms.
INCLUDE_UDMA	Include ultra-dma support for ide	Requires modifications to portkern.c and portio.c.
INCLUDE_82365_PCMCTRL	Include the 82365 PCMCIA controller	Requires modifications to portkern.c and portio.c.