Rtfs

RtfilesEx1.1 Application Notes

Revised June 2009

EBS Inc. 39 Court Street Groton MA 01450 USA
http://www.ebsembeddedsoftware.com

TABLE OF CONTENTS

Interface changes introduced in version 1.1.

Several changes were made to the external interfaces of RtfilesEx between the release of versions 1.1 and 1.0. This application note and some wrapper software are provided to help existing version 1.0 to upgrade to version 1.1.

The following major changes to the RtfilesEx interfaces have been made:

- The include directory has been moved.
  - In the previous release the include files were locate in rtfscommon\include\.
  - In the new release they are located in the directory include\.
- Some compile time configurations have been rearranged within, removed or moved in the rtfsconf.h, portconf.h and rtfsarch.h header files.
- The method for providing system wide buffer pools and assigning system configuration values like maximum number of files and maximum number of drives has changed.
  - In previous versions system wide parameters were set by changing constants a file named apicnfig.c and recompiling the library.
  - In the new version system wide parameters are set by filling in a structure that is passed to a project level callback function named rtfs_init_configuration().
- The method that provides per mount buffer pools and per mount configuration values, like the FAT cache size, and Failsafe buffer sizes, has changed.
  - In previous versions these values were configured when the device driver was attached and drive letters were assigned to the volume(s) on devices controlled by the driver.
  - In the new version, the device driver itself is responsible for attaching itself when media is inserted. It does this by calling pc_rtfs_media_insert() and passing in configuration values and the device handler entry points.
  - In the new version, volume wide parameters (there may be more than one volume inside multiple partitions on the media) are assigned through another callback function.

Upgrading from version 1.0.

- This application note provides instructions to help in the upgrade process.
- The release provides source code wrappers that will help migrate your configurations from version 1.0 and allow you to use version 1.0 device drivers.
  - The source code for this functionality is located in a few files in the directory rtfsdrivers\v10devwrap.
  - The device drivers provided in the rtfsdrivers directory are the same device drivers that were shipped with version 1.0, unchanged.
  - To enable this functionality the variable in rtfsconf.h named INCLUDE_V_1_0_DEVICES must be set to 1.

Upgrading header files.

Follow these notes to modify the new header files to mimic your old header files.

- The location of the header files has changed from rtfs\common\include\ to just include\. You must either update your project files or move the include subdirectory to its old location.
- The following headers have changed and must be modified.
- Rtfsarch.h
    - ***Do not copy your old file over this new one but edit the configurations to match.***
    - INCLUDE_DEBUG_TRUE_ASSERT was moved to rtfsarch.h from rtfsconf.h.
    - The following constants are new in rtfsarch.h - If enabled they provide useful features but by default they are left disabled. See the manual for more information.
        - INCLUDE_THREAD_SETENV_SUPPORT
        - INCLUDE_THREAD_EXIT_CALLBACK
        - RTFS_CACHE_LINE_SIZE_IN_BYTES
- Portconf.h
    - ***Do not copy your old file over this new one but edit the configurations to match.***
    - INCLUDE_V_1_0_DEVICES is enabled the device configuration for version 1.1 is provide by portconf.h and the configuration constants in portconf.h must be modified.
- Rtfsconf.h
    - ***Do not copy your old file over this new one but edit the configurations to match.***
    - INCLUDE_DEBUG_TRUE_ASSERT was moved to rtfsarch.h from rtfsconf.h
    - The following constants are still included in rtfsconf.h but they have been moved around within the file:
        - INCLUDE_BASIC_POSIX_EMULATION
        - INCLUDE_DEBUG_RUNTIME_STATS
        - INCLUDE_DEBUG_LEAK_CHECKING
        - INCLUDE_DEBUG_VERBOSE_ERRNO
        - INCLUDE_DEBUG_TEST_CODE
    - The following constants are obsolete in version 1.1 but they are still required for using version 1.0 interfaces with the 1.1 code base. They have been moved to the file rtfsdrivers\v10devwrapper\v10wrapper.h.
        - RTFS_CFG_LEAN
        - RTFS_CFG_SHARE_BUFFERS
        - RTFS_CFG_ALLOC_FROM_HEAP
        - RTFS_CFG_NUM_USERS
        - RTFS_CFG_DEFAULT_SECTOR_SIZE_BYTES
    - The following constants are new in version 1.1.
        - INCLUDE_NAND_DRIVER - The default is 1 but this should be set to 0 if you are not supporting nand.
        - INCLUDE_V_1_0_DEVICES - This must be set to 1 to support using version 1.0 interfaces with the 1.1 code base.
    - The following constants were removed in version 1.1.
        - INCLUDE_DYNAMIC_DRIVER

    - INCLUDE_DEBUG_SIM_ASSERT
    - INCLUDE_TELNET_TERMINAL
    - INCLUDE_SYS_TELNET_TERMINAL
    - STORE_DEVICE_NAMES_IN_DRIVE_STRUCT
    - RTFS_CFG_READONLY
- The following constant was moved from rtfsconf.h to the version1.1 application callback layer.
    - FAILSAFE_MODE_AUTOMATIC
- By default for version 1.1 the constants that determine what device drivers to build where moved from portconf.h to the bottom of rtfsconf.h.
    - But, if INCLUDE_V_1_0_DEVICES is enabled, these constants are redefined in portconf.h and the values in portconf.h, not rtfsconf.h determine if they are included or not.
        - INCLUDE_IDE,INCLUDE_PCMCIA,INCLUDE_PCMCIA_SRAM, INCLUDE_COMPACT_FLASH,INCLUDE_FLASH_FTL, INCLUDE_ROMDISK,INCLUDE_RAMDISK, INCLUDE_MMCCARD,INCLUDE_SMARTMEDIA, INCLUDE_FLOPPY,INCLUDE_HOSTDISK ,INCLUDE_WINDEV,INCLUDE_UDMA, INCLUDE_82365_PCMCTRL

Configuring version 1.1 using version 1.0 interfaces.

This section describes a set of files located the directory rtfsdrivers\v10devwrap. These files use similar interface to the files of the same name in version1.0.

- apicnfig.c
    - Provides a global configuration function using a mechanism similar to the method used in version 1.0.
        - Provides functionality previously provided by rtfscommon\source\apicnfig.c.
        - In version 1.1 and later this is done by a file named rtfsconfig.c in project directory.
        - When INCLUDE_V_1_0_DEVICES is set to 1 code in rtfsconfig.c is excluded and replaced by code in rtfsdrivers\v10devwrapper\apicnfig.c.
    - You may diff this file with your old version of apicnfig.c in rtfiles1.0\rtfscommon\source to determine what constants must be changed.
    - The files are similar in the section that contains defines configuration constants, from line 1 to around line 117.
    - Note: One configuration constant behaves quite differently.
        - The constant named "NBLKBUFFS" was previously used to configure the total number of directory buffers to be shared by all drives.
        - This is not what this value is used for now. NBLKBUFFS in this file now configures the value RTFS_CFG_MAX_SCRATCH_BUFFERS that was introduced in version 1.1. This needs to be set to a minimum of four.
        - The configuration previously provided by NBLKBUFFS is now provided by the constant

DEFAULT_NUM_BLOCK_BUFFERS_PER_DRIVE_PER_DRIVE in apirun.c.

- apirun.c
  - This file provides functionality previously provided by the file rtfscommon\source\apirun.c.
  - In version 1.1 and later this is done by a file named rtfsrun.c in application's project directory.
  - When INCLUDE_V_1_0_DEVICES is set to 1 the code in rtfsrun.c is excluded and replaced by code in rtfsdrivers\v10devwrapper\apirun.c.
  - When upgrading you should diff this file with your version of apirun.c in rtfiles1.0\rtfscommon\source to determine what constants must be changed.
  - The files are similar in the section that defines configuration constants, (between approximately lines 58 and 214).
  - Note: Version 1.1 introduces a new configuration constant called DEFAULT_INDEX_BUFFER_SIZE. This is the size in sectors of the buffer used by the Journaling package. The default setting is 1. With this setting Failsafe uses the same buffering algorithm as in version 1.0. Larger values will consume more memory but improve performance.
  - As noted in the previous section, one configuration constant behaves quite differently. The constant named "NBLKBUFFS" previously used to configure the total number of directory buffers to be shared by all drives. This is now the responsibility of the constant "DEFAULT_NUM_BLOCK_BUFFERS_PER_DRIVE_PER_DRIVE", defined in apirun.c. It follows the same model used by the fat buffers.
- apiinit.c
  - Provides a device driver attach and start functionality using a mechanism similar to the method used in version 1.0.
  - Provides functionality that was previously provided by rtfscommon\source\apiinit.c.
  - In version 1.1 and later, the attach and startup functionality by code in the file named rtfsrun.c in the application's project directory.
  - When INCLUDE_V_1_0_DEVICES is set, the version 1.1 attach code in rtfsrun.c is excluded and replaced by code in rtfsdrivers\v10devwrapper\apiinit.c.
  - When upgrading a version 1.0 configuration to version 1.1 you can to diff this file with your version of apiinit.c in rtfiles1.0\rtfscommon\source to determine what has changed.
  - Since apiinit.c is usually not changed much you may not see significant confiuration differences.
  - Differences you might detect are:
    - changes to drive letter assignments.
    - how many partitions are supported for device type
    - possibly io and interrupt address assignment.
  - Note: Apiinit.c passes certain paramaters like io address, interrupt number slot number to the device driver using fields in the drive structure. This mechanism is not supported for version 1.1 and later but except when using the INCLUDE_V_1_0_DEVICES methods.

Technical discussion.

Two additional file are supplied to complete the version 1.0 interface emulation.

- v10wrapper.h
  - This file is only included by 'C' files in the subdirectory rtfsdrivers\v10devwrap.
  - Contains shared definitions and several compile time definitions that were previously provided in rtfsconf.h.
  - The constants are: RTFS_CFG_LEAN, RTFS_CFG_READONLY, RTFS_CFG_SHARE_BUFFERS, RTFS_CFG_ALLOC_FROM_HEAP and RTFS_CFG_NUM_USERS.
  - To emulate an existing configuration set these constants to the values previously defined in rtfscommon\includes\rtfsconf.h.

- v10glue.c
  - ***You shouldn't have to modify any source code in this file but a description is provided here to help you understand the mechanism.***
  - This file contains the run time logic necessary to support using version 1.0 device drivers with version 1.1.
  - rtfs_poll_devices(void) - This is called by rtfsile1.1 every time the API is entered. It uses the method provided by rtfiles1.0 device drivers to detect media status changes and then uses the methods introduced in version 1.1 to process those changes.
    - In earlier versions device polling was done by the rtfs devio layer which called the device driver's DEVCTL_CHECKSTATUS service call to check if the driver has detected a device insert or remove.
    - The new model requires the device driver to detect media insertion or removal and call pc_rtfs_media_alert() when media is ejected and to call pc_rtfs_media_insert() when media is installed.
    - In rtfiles1.1 all of the configuration information needed for the media are provided as parameters to pc_rtfs_media_insert().
  - v1_0_insert_device() Is a helper fuction called by rtfs_poll_devices(void) when a device insert is detected. It uses the values that were provided by apirun.c to initialize a version 1.1 rtfs_media_insert_args parameter block and call pc_rtfs_media_insert().
  - v1_0_device_configure_media() - This is a version1.1 configure media callback function (called once per insertion) that provides configuration data to version 1.1 using data that was assigned in apiinit.c and apirun.c in version1.0.
  - v1_0_device_configure_volume() - This is a version1.1 configure media callback function that provides configuration data when a volume on the device is mounted. It returns data that was assigned by apiinit.c and apirun.c.
  - v1_0_device_ioctl() - This is a version1.1 device io control function. It provides a wrapper around the version 1.0 ioctl function.
  - v1_0_device_io() - This is a version1.1 device io (read/write) function. It provides a wrapper around version 1.0 device io function.

- pc_v1_0_diskio_configure() – This simulates the function of the same name that was provided in version1.0.
  - It is called by code in apirun.c.
  - It saves the configuration information in a structure similar to the configuration structure used in version1.0.
  - Values from these saved structures are later passed in a version1.1 compatible configuration block when the volume is mounted.
- v1_0_check_reserved_drives() – This and its companion function v1_0_release_reserved_drives() are used to support the version1.1 method that dynamically assigns drive structures when media is inserted while supporting version 1.0 device drivers requirement that drive structures must be reserved when the system is started and
- pc_calculate_chs() - This function is borrowed from version 1.0.
  - Version 1.1 device drivers are required to return valid h, c and n values when they they return a valid lba value.
  - Version 1.0 drivers are required to return 0 values in h, c and n when they they return a valid lba value.
  - This function is used to calculate valid values for h, c and n when a version 1.0 device driver returns an lba only response to the DEVCTL_GET_GEOMETRY ioctl() request.