

Re: [Ext2-devel] [PATCH 1/9] extents for ext4

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2006-08/msg04923.html>

- *From:* "Randy Dunlap" <rdunlap@xxxxxxxxxxxxx>
 - *Date:* Mon, 14 Aug 2006 11:05:28 -0700
-

On Mon, 14 Aug 2006 13:52:42 -0400 Jeff Garzik wrote:

IMO these non-CONFIG bits should just be ext4_

Agreed. Replacement patch below.

From: Randy Dunlap <rdunlap@xxxxxxxxxxxxx>

Rename ext3dev to ext4dev.

Signed-off-by: Randy Dunlap <rdunlap@xxxxxxxxxxxxx>

```
fs/Kconfig | 58 ++++++-----
fs/Makefile | 2 -
fs/ext4/Makefile | 10 +++----
fs/ext4/acl.h | 6 ++--
fs/ext4/file.c | 2 -
fs/ext4/inode.c | 2 -
fs/ext4/namei.c | 6 ++--
fs/ext4/super.c | 18 +++++-----
fs/ext4/symlink.c | 4 +--
fs/ext4/xattr.c | 8 +++--
fs/ext4/xattr.h | 8 +++--
include/linux/ext4_fs_i.h | 4 +--
12 files changed, 64 insertions(+), 64 deletions(-)
```

```
--- linux-2618-rc4-ext4.orig/fs/Kconfig
+++ linux-2618-rc4-ext4/fs/Kconfig
@@ -138,38 +138,38 @@ config EXT3_FS_SECURITY
If you are not using a security module that requires using
extended attributes for file security labels, say N.
```

```
-config EXT3DEV_FS
- tristate "Ext3dev/ext4 extended fs support development (EXPERIMENTAL)"
+config EXT4DEV_FS
+ tristate "Ext4dev extended fs support development (EXPERIMENTAL)"
```

Re: [Ext2-devel] [PATCH 1/9] extents for ext4

depends on EXPERIMENTAL

select JBD2

help

- Ext3dev is a predecessor filesystem of the next generation
- + Ext4dev is a predecessor filesystem of the next generation extended fs ext4, based on ext3 filesystem code. It will be
- renamed ext4 fs later, once ext3dev is mature and stabilized.
- + renamed ext4 fs later, once ext4dev is mature and stabilized.

Unlike the change from ext2 filesystem to ext3 filesystem,

- the on-disk format of ext3dev is not the same as ext3 any more:
- + the on-disk format of ext4dev is not the same as ext3 any more: it is based on extent maps and it supports 48-bit physical block numbers. These combined on-disk format changes will allow
- ext3dev/ext4 to handle more than 16 TB filesystem volumes ---
- + ext4dev to handle more than 16 TB filesystem volumes --- a hard limit that ext3 cannot overcome without changing the on-disk format.

- Other than extent maps and 48-bit block numbers, ext3dev also is
- + Other than extent maps and 48-bit block numbers, ext4dev also is likely to have other new features such as persistent preallocation, high resolution time stamps, and larger file support etc. These
- features will be added to ext3dev gradually.
- + features will be added to ext4dev gradually.

To compile this file system support as a module, choose M here. The

- module will be called ext3dev. Be aware, however, that the filesystem
- + module will be called ext4dev. Be aware, however, that the filesystem of your root partition (the one containing the directory /) cannot be compiled as a module, and so this could be dangerous.

If unsure, say N.

-config EXT3DEV_FS_XATTR

- bool "Ext3dev extended attributes"

- depends on EXT3DEV_FS

+config EXT4DEV_FS_XATTR

+ bool "Ext4dev extended attributes"

+ depends on EXT4DEV_FS

default y

help

Extended attributes are name:value pairs associated with inodes by

@@ -178,11 +178,11 @@ config EXT3DEV_FS_XATTR

If unsure, say N.

- You need this for POSIX ACL support on ext3dev/ext4.

+ You need this for POSIX ACL support on ext4dev.

-config EXT3DEV_FS_POSIX_ACL

Re: [Ext2-devel] [PATCH 1/9] extents for ext4

Re: [Ext2-devel] [PATCH 1/9] extents for ext4

- bool "Ext3dev POSIX Access Control Lists"
- depends on EXT3DEV_FS_XATTR
- +config EXT4DEV_FS_POSIX_ACL
- + bool "Ext4dev POSIX Access Control Lists"
- + depends on EXT4DEV_FS_XATTR
- select FS_POSIX_ACL

help
POSIX Access Control Lists (ACLs) support permissions for users and
@@ -193,14 +193,14 @@ config EXT3DEV_FS_POSIX_ACL

If you don't know what Access Control Lists are, say N

- config EXT3DEV_FS_SECURITY
- bool "Ext3dev Security Labels"
- depends on EXT3DEV_FS_XATTR
- +config EXT4DEV_FS_SECURITY
- + bool "Ext4dev Security Labels"
- + depends on EXT4DEV_FS_XATTR

help
Security labels support alternative access control models
implemented by security modules like SELinux. This option
enables an extended attribute handler for file security

- labels in the ext3dev/ext4 filesystem.
- + labels in the ext4dev filesystem.

If you are not using a security module that requires using
extended attributes for file security labels, say N.

@@ -242,22 +242,22 @@ config JBD2

help
This is a generic journaling layer for block devices that support
both 32-bit and 64-bit block numbers. It is currently used by

- the ext3dev/ext4 filesystem, but it could also be used to add
- + the ext4dev filesystem, but it could also be used to add

journal support to other file systems or block devices such
as RAID or LVM.

- If you are using ext3dev/ext4, you need to say Y here. If you are not
- using ext3dev/ext4 then you will probably want to say N.
- + If you are using ext4dev, you need to say Y here. If you are not
- + using ext4dev then you will probably want to say N.

To compile this device as a module, choose M here. The module will be

- called jbd2. If you are compiling ext3dev/ext4 into the kernel,
- + called jbd2. If you are compiling ext4dev into the kernel,

you cannot compile this code as a module.

config JBD2_DEBUG

- bool "JBD2 (ext3dev/ext4) debugging support"
- + bool "JBD2 (ext4dev) debugging support"

depends on JBD2

help

Re: [Ext2-devel] [PATCH 1/9] extents for ext4

```
- If you are using the ext3dev/ext4 journaled file system (or
+ If you are using the ext4dev journaled file system (or
potentially any other filesystem/device using JBD2), this option
allows you to enable debugging output while the system is running,
in order to help track down any problems you are having.
@@ -272,9 +272,9 @@ config JBD2_DEBUG
config FS_MBCACHE
# Meta block cache for Extended Attributes (ext2/ext3/ext4)
tristate
- depends on EXT2_FS_XATTR || EXT3_FS_XATTR || EXT3DEV_FS_XATTR
- default y if EXT2_FS=y || EXT3_FS=y || EXT3DEV_FS=y
- default m if EXT2_FS=m || EXT3_FS=m || EXT3DEV_FS=m
+ depends on EXT2_FS_XATTR || EXT3_FS_XATTR || EXT4DEV_FS_XATTR
+ default y if EXT2_FS=y || EXT3_FS=y || EXT4DEV_FS=y
+ default m if EXT2_FS=m || EXT3_FS=m || EXT4DEV_FS=m

config REISERFS_FS
tristate "Reiserfs support"
--- linux-2618-rc4-ext4.orig/fs/Makefile
+++ linux-2618-rc4-ext4/fs/Makefile
@@ -54,7 +54,7 @@ obj-$(CONFIG_PROFILING) += dcookies.o
# Do not add any filesystems before this line
obj-$(CONFIG_REISERFS_FS) += reiserfs/
obj-$(CONFIG_EXT3_FS) += ext3/ # Before ext2 so root fs can be ext3
-obj-$(CONFIG_EXT3DEV_FS) += ext4/ # Before ext2 so root fs can be ext3dev
+obj-$(CONFIG_EXT4DEV_FS) += ext4/ # Before ext2 so root fs can be ext3dev
obj-$(CONFIG_JBD) += jbd/
obj-$(CONFIG_JBD2) += jbd2/
obj-$(CONFIG_EXT2_FS) += ext2/
--- linux-2618-rc4-ext4.orig/fs/ext4/Makefile
+++ linux-2618-rc4-ext4/fs/ext4/Makefile
@@ -2,11 +2,11 @@
# Makefile for the linux ext4-filesystem routines.
#

-obj-$(CONFIG_EXT3DEV_FS) += ext3dev.o
+obj-$(CONFIG_EXT4DEV_FS) += ext4dev.o

-ext3dev-y := balloc.o bitmap.o dir.o file.o fsync.o ialloc.o inode.o \
+ext4dev-y := balloc.o bitmap.o dir.o file.o fsync.o ialloc.o inode.o \
ioctl.o namei.o super.o symlink.o hash.o resize.o extents.o

-ext3dev-$(CONFIG_EXT3DEV_FS_XATTR) += xattr.o xattr_user.o xattr_trusted.o
-ext3dev-$(CONFIG_EXT3DEV_FS_POSIX_ACL) += acl.o
-ext3dev-$(CONFIG_EXT3DEV_FS_SECURITY) += xattr_security.o
+ext4dev-$(CONFIG_EXT4DEV_FS_XATTR) += xattr.o xattr_user.o xattr_trusted.o
+ext4dev-$(CONFIG_EXT4DEV_FS_POSIX_ACL) += acl.o
+ext4dev-$(CONFIG_EXT4DEV_FS_SECURITY) += xattr_security.o
--- linux-2618-rc4-ext4.orig/fs/ext4/acl.h
+++ linux-2618-rc4-ext4/fs/ext4/acl.h
@@ -51,7 +51,7 @@ static inline int ext4_acl_count(size_t
```

```

}
}

#ifndef CONFIG_EXT3DEV_FS_POSIX_ACL
#ifdef CONFIG_EXT4DEV_FS_POSIX_ACL

/* Value for inode->u.ext4_i.i_acl and inode->u.ext4_i.i_default_acl
if the ACL has not been cached */
@@ -62,7 +62,7 @@ extern int ext4_permission (struct inode
extern int ext4_acl_chmod (struct inode *);
extern int ext4_init_acl (handle_t *, struct inode *, struct inode *);

#else /* CONFIG_EXT3DEV_FS_POSIX_ACL */
+else /* CONFIG_EXT4DEV_FS_POSIX_ACL */
#include <linux/sched.h>
#define ext4_permission NULL

@@ -77,5 +77,5 @@ ext4_init_acl(handle_t *handle, struct i
{
return 0;
}
#endif /* CONFIG_EXT3DEV_FS_POSIX_ACL */
+endif /* CONFIG_EXT4DEV_FS_POSIX_ACL */

--- linux-2618-rc4-ext4.orig/fs/ext4/xattr.h
+++ linux-2618-rc4-ext4/fs/ext4/xattr.h
@@ -56,7 +56,7 @@ struct ext4_xattr_entry {
#define EXT4_XATTR_SIZE(size) \
(((size) + EXT4_XATTR_ROUND) & ~EXT4_XATTR_ROUND)

#ifndef CONFIG_EXT3DEV_FS_XATTR
+ifdef CONFIG_EXT4DEV_FS_XATTR

extern struct xattr_handler ext4_xattr_user_handler;
extern struct xattr_handler ext4_xattr_trusted_handler;
@@ -79,7 +79,7 @@ extern void exit_ext4_xattr(void);

extern struct xattr_handler *ext4_xattr_handlers[];

#ifndef CONFIG_EXT3DEV_FS_XATTR
+else /* CONFIG_EXT4DEV_FS_XATTR */

static inline int
ext4_xattr_get(struct inode *inode, int name_index, const char *name,
@@ -131,9 +131,9 @@ exit_ext4_xattr(void)

#define ext4_xattr_handlers NULL

#ifndef CONFIG_EXT3DEV_FS_XATTR */
+endif /* CONFIG_EXT4DEV_FS_XATTR */

```

Re: [Ext2-devel] [PATCH 1/9] extents for ext4

```

-#ifdef CONFIG_EXT3DEV_FS_SECURITY
+#ifdef CONFIG_EXT4DEV_FS_SECURITY
extern int ext4_init_security(handle_t *handle, struct inode *inode,
struct inode *dir);
#else
--- linux-2618-rc4-ext4.orig/fs/ext4/file.c
+++ linux-2618-rc4-ext4/fs/ext4/file.c
@@ -126,7 +126,7 @@ const struct file_operations ext4_file_o
struct inode_operations ext4_file_inode_operations = {
.truncate = ext4_truncate,
.setattr = ext4_setattr,
-#ifdef CONFIG_EXT3DEV_FS_XATTR
+#ifdef CONFIG_EXT4DEV_FS_XATTR
.setxattr = generic_setxattr,
.getxattr = generic_getxattr,
.listxattr = ext4_listxattr,
--- linux-2618-rc4-ext4.orig/fs/ext4/inode.c
+++ linux-2618-rc4-ext4/fs/ext4/inode.c
@@ -2589,7 +2589,7 @@ void ext4_read_inode(struct inode * inod
struct buffer_head *bh;
int block;

-#ifdef CONFIG_EXT3DEV_FS_POSIX_ACL
+#ifdef CONFIG_EXT4DEV_FS_POSIX_ACL
ei->i_acl = EXT4_ACL_NOT_CACHED;
ei->i_default_acl = EXT4_ACL_NOT_CACHED;
#endif
--- linux-2618-rc4-ext4.orig/fs/ext4/namei.c
+++ linux-2618-rc4-ext4/fs/ext4/namei.c
@@ -1689,7 +1689,7 @@ retry:
err = PTR_ERR(inode);
if (!IS_ERR(inode)) {
init_special_inode(inode, inode->i_mode, rdev);
-#ifdef CONFIG_EXT3DEV_FS_XATTR
+#ifdef CONFIG_EXT4DEV_FS_XATTR
inode->i_op = &ext4_special_inode_operations;
#endif
err = ext4_add_nondir(handle, dentry, inode);
@@ -2364,7 +2364,7 @@ struct inode_operations ext4_dir_inode_o
.mknod = ext4_mknod,
.rename = ext4_rename,
.setattr = ext4_setattr,
-#ifdef CONFIG_EXT3DEV_FS_XATTR
+#ifdef CONFIG_EXT4DEV_FS_XATTR
.setxattr = generic_setxattr,
.getxattr = generic_getxattr,
.listxattr = ext4_listxattr,
@@ -2375,7 +2375,7 @@ struct inode_operations ext4_dir_inode_o

struct inode_operations ext4_special_inode_operations = {
.setattr = ext4_setattr,

```

Re: [Ext2-devel] [PATCH 1/9] extents for ext4

```

-#ifdef CONFIG_EXT3DEV_FS_XATTR
+#ifdef CONFIG_EXT4DEV_FS_XATTR
.setxattr = generic_setxattr,
.getxattr = generic_getxattr,
.listxattr = ext4_listxattr,
--- linux-2618-rc4-ext4.orig/fs/ext4/super.c
+++ linux-2618-rc4-ext4/fs/ext4/super.c
@@ -448,7 +448,7 @@ static struct inode *ext4_alloc_inode(st
ei = kmem_cache_alloc(ext4_inode_cache, SLAB_NOFS);
if (!ei)
return NULL;
-#ifdef CONFIG_EXT3DEV_FS_POSIX_ACL
+#ifdef CONFIG_EXT4DEV_FS_POSIX_ACL
ei->i_acl = EXT4_ACL_NOT_CACHED;
ei->i_default_acl = EXT4_ACL_NOT_CACHED;
#endif
@@ -470,7 +470,7 @@ static void init_once(void * foo, kmem_c
if ((flags & (SLAB_CTOR_VERIFY|SLAB_CTOR_CONSTRUCTOR)) ==
SLAB_CTOR_CONSTRUCTOR) {
INIT_LIST_HEAD(&ei->i_orphan);
-#ifdef CONFIG_EXT3DEV_FS_XATTR
+#ifdef CONFIG_EXT4DEV_FS_XATTR
init_rwsem(&ei->xattr_sem);
#endif
mutex_init(&ei->truncate_mutex);
@@ -499,7 +499,7 @@ static void destroy_inodecache(void)
static void ext4_clear_inode(struct inode
*inode)
{
struct ext4_block_alloc_info *rsv = EXT4_I(inode)->i_block_alloc_info;
-#ifdef CONFIG_EXT3DEV_FS_POSIX_ACL
+#ifdef CONFIG_EXT4DEV_FS_POSIX_ACL
if (EXT4_I(inode)->i_acl &&
EXT4_I(inode)->i_acl != EXT4_ACL_NOT_CACHED) {
posix_acl_release(EXT4_I(inode)->i_acl);
@@ -793,7 +793,7 @@ static int parse_options (char *options,
case Opt_orlov:
clear_opt (sbi->s_mount_opt, OLDALLOC);
break;
-#ifdef CONFIG_EXT3DEV_FS_XATTR
+#ifdef CONFIG_EXT4DEV_FS_XATTR
case Opt_user_xattr:
set_opt (sbi->s_mount_opt, XATTR_USER);
break;
@@ -806,7 +806,7 @@ static int parse_options (char *options,
printk("EXT4 (no)user_xattr options not supported\n");
break;
#endif
-#ifdef CONFIG_EXT3DEV_FS_POSIX_ACL
+#ifdef CONFIG_EXT4DEV_FS_POSIX_ACL
case Opt_acl:
set_opt(sbi->s_mount_opt, POSIX_ACL);

```

Re: [Ext2-devel] [PATCH 1/9] extents for ext4

```
break;
@@ -2683,9 +2683,9 @@ static int ext4_get_sb(struct file_syste
return get_sb_bdev(fs_type, flags, dev_name, data, ext4_fill_super, mnt);
}
```

```
-static struct file_system_type ext3dev_fs_type = {
+static struct file_system_type ext4_fs_type = {
.owner = THIS_MODULE,
- .name = "ext3dev",
+ .name = "ext4dev",
.get_sb = ext4_get_sb,
.kill_sb = kill_block_super,
.fs_flags = FS_REQUIRES_DEV,
@@ -2699,7 +2699,7 @@ static int __init init_ext4_fs(void)
err = init_inodecache();
if (err)
goto out1;
- err = register_filesystem(&ext3dev_fs_type);
+ err = register_filesystem(&ext4_fs_type);
if (err)
goto out;
return 0;
@@ -2712,7 +2712,7 @@ out1:
```

```
static void __exit exit_ext4_fs(void)
{
- unregister_filesystem(&ext3dev_fs_type);
+ unregister_filesystem(&ext4_fs_type);
destroy_inodecache();
exit_ext4_xattr();
}
--- linux-2618-rc4-ext4.orig/fs/ext4/symlink.c
+++ linux-2618-rc4-ext4/fs/ext4/symlink.c
@@ -34,7 +34,7 @@ struct inode_operations ext4_symlink_ino
.readlink = generic_readlink,
.follow_link = page_follow_link_light,
.put_link = page_put_link,
-#ifdef CONFIG_EXT3DEV_FS_XATTR
+#ifdef CONFIG_EXT4DEV_FS_XATTR
.setxattr = generic_setxattr,
.getxattr = generic_getxattr,
.listxattr = ext4_listxattr,
@@ -45,7 +45,7 @@ struct inode_operations ext4_symlink_ino
struct inode_operations ext4_fast_symlink_inode_operations = {
.readlink = generic_readlink,
.follow_link = ext4_follow_link,
-#ifdef CONFIG_EXT3DEV_FS_XATTR
+#ifdef CONFIG_EXT4DEV_FS_XATTR
.setxattr = generic_setxattr,
.getxattr = generic_getxattr,
.listxattr = ext4_listxattr,
```

Re: [Ext2-devel] [PATCH 1/9] extents for ext4

```
--- linux-2618-rc4-ext4.orig/fs/ext4/xattr.c
+++ linux-2618-rc4-ext4/fs/ext4/xattr.c
@@ -104,12 +104,12 @@ static struct mb_cache *ext4_xattr_cache

static struct xattr_handler *ext4_xattr_handler_map[] = {
[EXT4_XATTR_INDEX_USER] = &ext4_xattr_user_handler,
-#ifdef CONFIG_EXT3DEV_FS_POSIX_ACL
+#ifdef CONFIG_EXT4DEV_FS_POSIX_ACL
[EXT4_XATTR_INDEX_POSIX_ACL_ACCESS] = &ext4_xattr_acl_access_handler,
[EXT4_XATTR_INDEX_POSIX_ACL_DEFAULT] = &ext4_xattr_acl_default_handler,
#endif
[EXT4_XATTR_INDEX_TRUSTED] = &ext4_xattr_trusted_handler,
-#ifdef CONFIG_EXT3DEV_FS_SECURITY
+#ifdef CONFIG_EXT4DEV_FS_SECURITY
[EXT4_XATTR_INDEX_SECURITY] = &ext4_xattr_security_handler,
#endif
};
@@ -117,11 +117,11 @@ static struct xattr_handler *ext4_xattr_
struct xattr_handler *ext4_xattr_handlers[] = {
&ext4_xattr_user_handler,
&ext4_xattr_trusted_handler,
-#ifdef CONFIG_EXT3DEV_FS_POSIX_ACL
+#ifdef CONFIG_EXT4DEV_FS_POSIX_ACL
&ext4_xattr_acl_access_handler,
&ext4_xattr_acl_default_handler,
#endif
-#ifdef CONFIG_EXT3DEV_FS_SECURITY
+#ifdef CONFIG_EXT4DEV_FS_SECURITY
&ext4_xattr_security_handler,
#endif
NULL
--- linux-2618-rc4-ext4.orig/include/linux/ext4_fs_i.h
+++ linux-2618-rc4-ext4/include/linux/ext4_fs_i.h
@@ -103,7 +103,7 @@ struct ext4_inode_info {
struct ext4_block_alloc_info *i_block_alloc_info;

__u32 i_dir_start_lookup;
-#ifdef CONFIG_EXT3DEV_FS_XATTR
+#ifdef CONFIG_EXT4DEV_FS_XATTR
/*
* Extended attributes can be read independently of the main file
* data. Taking i_mutex even when reading would cause contention
@@ -113,7 +113,7 @@ struct ext4_inode_info {
*/
struct rw_semaphore xattr_sem;
#endif
-#ifdef CONFIG_EXT3DEV_FS_POSIX_ACL
+#ifdef CONFIG_EXT4DEV_FS_POSIX_ACL
struct posix_acl *i_acl;
struct posix_acl *i_default_acl;
#endif
```

Re: [Ext2-devel] [PATCH 1/9] extents for ext4

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@xxxxxxxxxxxxxxxxx

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>