

---

Subject: [PATCH 1/6][NET-2.6.24] Introduce the seq\_open\_private()  
Posted by [Pavel Emelianov](#) on Tue, 09 Oct 2007 15:52:58 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

This function allocates the zeroed chunk of memory and call seq\_open(). The \_\_seq\_open\_private() helper returns the allocated memory to make it possible for the caller to initialize it.

Signed-off-by: Pavel Emelianov <xemul@openvz.org>

---

```
diff --git a/fs/seq_file.c b/fs/seq_file.c
index bbb19be..ca71c11 100644
--- a/fs/seq_file.c
+++ b/fs/seq_file.c
@@ -429,6 +429,39 @@ int seq_release_private(struct inode *in
 }
EXPORT_SYMBOL(seq_release_private);

+void *__seq_open_private(struct file *f, const struct seq_operations *ops,
+ int psize)
+{
+ int rc;
+ void *private;
+ struct seq_file *seq;
+
+ private = kzalloc(psize, GFP_KERNEL);
+ if (private == NULL)
+ goto out;
+
+ rc = seq_open(f, ops);
+ if (rc < 0)
+ goto out_free;
+
+ seq = f->private_data;
+ seq->private = private;
+ return private;
+
+out_free:
+ kfree(private);
+out:
+ return NULL;
+}
+EXPORT_SYMBOL(__seq_open_private);
+
+int seq_open_private(struct file *filp, const struct seq_operations *ops,
```

```

+ int psize)
+{
+ return __seq_open_private(filp, ops, psize) ? 0 : -ENOMEM;
+}
+EXPORT_SYMBOL(seq_open_private);
+
+int seq_putc(struct seq_file *m, char c)
+{
+    if (m->count < m->size) {
diff --git a/include/linux/seq_file.h b/include/linux/seq_file.h
index 83783ab..8bf1e05 100644
--- a/include/linux/seq_file.h
+++ b/include/linux/seq_file.h
@@ -46,6 +46,8 @@ int seq_path(struct seq_file *, struct v

int single_open(struct file *, int (*)(struct seq_file *, void *), void *);
int single_release(struct inode *, struct file *);
+void *__seq_open_private(struct file *, const struct seq_operations *, int);
+int seq_open_private(struct file *, const struct seq_operations *, int);
int seq_release_private(struct inode *, struct file *);

#define SEQ_START_TOKEN ((void *)1)

```

---