

NAME

unixfile – creating and destroying UnixFile objects

SYNOPSIS

```

VASResult      shore_vas::mkUnixFile(
    const Path  name,
    mode_t      mode,
    const vec_t &contents,    // -- initial value
    lrid_t      *result
);

VASResult      shore_vas::rmUnixFile(
    const lrid_t &dir,        // -- can't be root("/")
    const Path  name
);

```

DESCRIPTION

Value-added servers can create and destroy UnixFile objects with these methods. UnixFile objects are Shore objects with a single attribute, of type TEXT.

ARGUMENTS

MkUnixFile creates an object with the path or file name given in *name*. The protection bits are given in *mode*. The object is initialized with the data given in *contents*. The SVAS writes logical object identifier of the new object into the space to which *result* points. If *result* is null, the resulting object identifier is not written.

RmUnixFile removes the object named *name* in the directory whose object identifier is *dir*.

ENVIRONMENT

These methods are available only to value-added servers.

These methods can be used only in a transaction.

ERRORS

Deadlocks can occur while locks are being acquired. See **transaction(svas)** for information about deadlocks.

A complete list of errors is in **errors(svas)**.

VERSION

This manual page applies to Version 1.1 of the Shore software.

SPONSORSHIP

The Shore project is sponsored by the Advanced Research Project Agency, ARPA order number 018 (formerly 8230), monitored by the U.S. Army Research Laboratory under contract DAAB07-91-C-Q518.

COPYRIGHT

Copyright © 1994, 1995, 1996, 1997, Computer Sciences Department, University of WisconsinMadison. All Rights Reserved.

SEE ALSO

errors(svas), **transaction(svas)**, and **text(svas)**.