NAME

sysprops – system properties of Shore Objects

SYNOPSIS

```
typedef serial_t_guts   Ref;
typedef  recsize_t      ObjectOffset;
typedef  recsize_t      ObjectSize;


struct AnonProps {
    Ref     pool;  /* Ref<Pool> or Ref<Module> (same volume) */
};
struct RegProps {
    short   nlink;
    mode_t  mode;
    uid_t   uid;
    gid_t   gid;
    time_t  atime;
    time_t  mtime;
    time_t  ctime;
};

struct SysProps {
    lvid_t          volume; /* like device */
    Ref             ref;    /* like inode */
    Ref             type;
    ObjectSize      csize;
    ObjectSize      hsize;
    ObjectOffset    tstart;
    int             nindex;
    ObjectKind      tag;
    union    {
    // case  KindAnonymous :
        AnonProps ap;

    // case  KindRegistered :
        RegProps rp;

    // case  KindTransient :

    } specific_u;
};

// for registered objects:
VASResult       shore_vas::sysprops(
    const Path          name,   // Shore path --
    SysProps            *sysprops
);

// for anonymous objects
// or registered objects when not
// following symlinks or cross-references
// in the path lookup
VASResult       shore_vas::sysprops(
    const lrid_t         &loid,
    SysProps             *sysprops,
    bool          wholepage=FALSE,
    LockMode             lock=S,
    bool          *is_unix_file=NULL,
    bool          *page_cached=NULL
);
```

DESCRIPTION

The type **SysProps** describes the properties of every Shore object. It is Shore's analog of `struct stat`.

The functions **sysprops** retrieve the system properties of an object, in the form of a *sysprops* structure.

The contents of the structure have the following meanings:

volume
The logical volume identifier for the volume on which the object resides.

ref The serial number of the object. Volume and ref, together form the logical OID of the object.

csize The size of the object's core. This is fixed.

hsize The size of the object's heap.

tstart The offset, in bytes, from the start of the object, of the object's **TEXT** attribute.

nindex
The number of manual indexes that this object contains. (This reflects only the number that the Language Binding said the object would contain. It does not reflect the number of manual indexes that have actually been added to the object. There is no nice way to tell how many indexes have been added to the object; it is assumed that the transaction that creates the object also creates the indexes, or has its own mechanism, for managing the indexes attached to an object.)

tag Indicates if the object is anonymous, **anonymous, registered,** or **transient.**

specific_u.ap.pool
If the object is anonymous, this is the serial number of the object's pool.

specific_u.rp.nlink
The number of hard (directory) links to the object.

specific_u.rp.mode
The number of hard (directory) links to the object.

specific_u.rp.uid
The user ID of the owner of the object.

specific_u.rp.gid
The group ID of the group of the object.

specific_u.rp.atime
The access time of the object (set by **utimes).**

specific_u.rp.mtime
The modification time of the object (set by **utimes** and by updates to the object).

specific_u.rp.ctime
The status-change time of the object (set by **utimes, chMod, chOwn,** and **chGrp).**

ARGUMENTS

In the first form, which is used for **registered objects,** the argument *name* is a pathname for the object. A **share lock** is acquired for each component directory in the path.

The second form is used for registered objects when their OIDs are known, and for **anonymous objects.** This form of **sysprops** has three arguments with default values.

Wholepage, if TRUE, requests the SVAS to bring an entire page of objects into the client's address space, if it can. This happens if and only if the requested object is anonymous, is a **small object,** and the caller is the client (not another value-added server). If the entire page is copied, the entire page is locked.

The argument *lock* can cause the SVAS to acquire a lock stronger lock than the default share lock on the object (or page of objects).

The argument *is_unix_file* is a pointer to a (pre-allocated) Boolean variable in the caller's address space. If the pointer is non-NULL, the SVAS returns TRUE if the object has a **TEXT** attribute.

In both forms, the system properties are written into the caller's address space at the location to which the argument *sysprops* points. This space must have been allocated by the caller prior to calling the function.

The argument *page_cached* is an (optional) output argument into which the SVAS writes *true* if the function resulted in a page of small (anonymous) objects being cached. See **page(svas)** for information about how to use the page of cached objects.

ENVIRONMENT

Both functions are available in both the server and client environments, but for servers, the argument *wholepage* is ignored.

Sysprops can be used only in a transaction.

ERRORS

The return values are valid only if **sysprops** returns ST_OK.

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

SEE ALSO

permissions(svas), errors(svas), text(svas), file_system(svas), lockObj(svas), pool(svas), object(svas), utimes(svas), and **transaction(svas).**