NAME

stat – retrieve information about an object

SYNOPSIS

```
#include <ShoreApp.h>
shrc Shore::stat(const char *path, OStat *osp);
```

DESCRIPTION

The **stat** method retrieves information about the object indicated by *path*, and fills in the structure given by *osp*. If *path* specifies a symbolic link or a cross-reference, this method will retrieve information about the object referenced by the link or cross-reference, not the link or cross-reference object. An alternate way of retrieving this information for an object of a known type is through the language binding (see **ref(cxxlb)**).

The *OStat* structure, defined in OCTypes.h, actually consists of three types. *OStat* contains some fields that are valid for all objects, registered or anonymous. AnonStat has fields that are valid only for anonymous objects, and RegStat has fields that are valid only for registered objects. The *kind* field of *OStat* indicates whether an object is registered or anonymous. The structures are defined as follows:

```
struct AnonStat
{
    LOID pool;          // loid of pool in which obj resides
};

struct RegStat
{
    short nlink;        // num. dir. entries pointing to object
    mode_t mode;        // permissions flags
    uid_t uid;          // owner
    gid_t gid;          // group
    time_t atime;       // access time
    time_t mtime;       // modify time
    time_t ctime;       // props change time
};

struct OStat
{
    LOID loid;          // object’s logical oid
    LOID type_loid;     // loid of object’s type object
    ObjectSize csize;   // core size
    ObjectSize hsize;   // heap size
    int nindices;       // number of indices in object
    ObjectKind kind;    // KindRegistered or KindAnonymous
    AnonStat astat;     // if kind == KindAnonymous
    RegStat  rstat;     // if kind == KindRegistered
};
```

The values of the *kind* field are defined in vas_types.h. Although five values are defined in that file, only two of those values (KindRegistered and KindAnonymous) can be found in the *kind* field in the current implementation.

Stat obtains a SH-mode (share-mode) lock on the object.

BUGS

Changes to the heap size of an object that occurred during the current transaction might not be reflected immediately in the *hsize* field of the OStat structure. Typically, the reported value of *hsize* will be zero for objects created within the transaction. For objects created in earlier transactions, the reported heap size will typically be last committed value.

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

ostat(cxxlb).