NAME

transaction – making sequences of Shore operations atomic

SYNOPSIS

```
VASResult    shore_vas::trans(tid_t    *tid=0);

VASResult    shore_vas::beginTrans( int        degree,
                                    tid_t      *tid = 0);

VASResult    shore_vas::abortTrans(int  reason = ST_UserAbort);
VASResult    shore_vas::abortTrans(const  tid_t    &tid,
                                   int     reason = ST_UserAbort);

VASResult    shore_vas::commitTrans();
VASResult    shore_vas::commitTrans(const  tid_t    &tid);
```

DESCRIPTION

Trans returns ST_FAILURE if no transaction is associated with the caller. If a transaction is associated with the caller, whatever its state, **trans** returns ST_OK. If *tid* is non-null, the transaction identifier for the transaction is returned in its target.

BeginTrans, abortTrans, and **commitTrans** start, roll back, and commit transactions, respectively.

ARGUMENTS

If the argument *tid* to **beginTrans** and **trans** is non-null, the transaction identifier is written into the area addressed by *tid.*

For **commitTrans** and **abortTrans,** the argument *tid* identifies the transaction to be committed or aborted, respectively. If no such argument is given, the **current** transaction is committed or aborted.

BeginTrans takes an argument, *degree,* which takes a value between 0 and 3, respectively.

DEGREE

The degree refers to the ACID (atomicity, consistency isolation, durability) properties of the transaction. See *Gray, J., Reuter, A. Transaction Processing: concepts and techniques, 1993.* for details. **Degree 0, 1, and 2 transactions are not implemented.**

ENVIRONMENT

All these methods are available on both the server and clients.

ERRORS

In general, user errors in SVAS methods cause the transaction to be rolled back to its state when the SVAS method was called. Some errors are serious enough to cause the entire transaction to be aborted.

After a method returns in error (it returns SVAS_FAILURE), the *shore_vas* object's *status* indicates *that* the reason for the failure is ST_SmFailure. (the Storage Manager detected a problem; the Storage Manager's reason indicates that a deadlock occurred). The status also indicates whether the transaction was aborted or is still active. If the transaction was aborted, The caller must recognize the condition and start a new transaction in order to proceed. to be aborted.

REFERENCES

Gray, J., Reuter, A. Transaction Processing: concepts and techniques, 1993.

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

sysprops(svas), writeObj(svas), truncObj(svas), appendObj(svas), lockObj(svas), errors(svas), and **transaction(svas).**