Using the EXODUS Storage Manager V3.0

Differences between V2 and V3.0.

Last revision: May, 1993

1. Introduction

This document summarizes the differences between versions 2 and 3.0 of the Exodus Storage Manager. “V2” refers to all Version 2 releases: V2.0, V2.1, and V2.2). This document includes references to the sections of the V3.0 user guide, **Using the EXODUS Storage Manager V3.0**. The section numbers do not correspond to the V2 documents. The reader is assumed to be familiar with a V2 release.

2. Release

(1) V3.0 runs on HP 7xx platforms running under HP-UX.

(2) The makefiles have been changed so that the sources are combined before making; V3.0 compiles in substantially less time than V2, and it uses less disk space for object files.

(3) V3.0 is made with g++ (V2 was made with ATT C++).

3. General Functionality

(1) Includes bulk-loading of linear hashed indexes. (Section 4.10.3)

(2) Includes two forms of distributed transactions. “Internally distributed” transactions are transactions that span several Storage Manager servers. The application - client library interface supports these with no changes from V2. An “externally distributed” transaction is a Storage Manager transaction (possibly using several servers) that is a “thread” of a “global transaction” that may involve other transaction processing systems. The application - client library interface supports these transactions with a set of new functions. (Sections 4.3, 4.11)

(3) The application no longer uses file names for volumes. Volumes are known by their volume IDs. Only servers know how to map a volume ID to a Unix path name. (Sections 4.4, 5.1.)

(4) Volumes are “auto-mounted”: applications no longer explicitly open and close connections to servers. References to data cause the proper servers to be contacted. (Sections 4.4, 5.1.)

(5) The servers and client library keep expanded statistics about their message traffic. (Section 4.11.2)

4. Application - Client Library Interface

4.1. Reporting Errors

(1) The client library exports a new global variable, `sm_reason`. When a transaction is aborted by the Storage Manager, `sm_reason` indicates why the transaction was aborted. (Section 4.1)

(2) V3.0 is a little smarter about reporting that a transaction was aborted. In some cases when V2 reported an “unknown transaction identifier” error because it lost track of the transaction that was aborted, V3.0 reports “transaction aborted”. (Section 4.1)

4.2. New Functions

(1) New administrative functions allow a client to determine the type of a volume and print the expanded message statistics. (Section 4.11.2)

(2) New administrative functions allow a client to change the server’s table of mountable volumes (a table that did not exist in V2). (Section 4.11.2)

(3) sm_RemoveRootEntry( ) allows root entries to be removed. (Section 4.5.)

4.3. Changed Functions

(1) The function sm_Initialize( ) no longer takes any parameters. (Section 4.2)

(2) All the administrative functions have new arguments to determine which server(s) are of interest. (Section 4.11.2)

(3) The arguments to the function sm_MountVolume( ) have new meanings, although the function’s prototype is unchanged for the sake of backward compatibility. (Section 4.4)

(4) The sm_* functions (Sections 4.7.1, 4.7.3) that create objects or change the sizes of objects break their operations down into chunks of a “reasonable” size when the operation involves a large amount of data. These functions take into account the size of the buffer group when chosing a reasonable size. As a result, operations are less likely to fail for lack of buffer space.

4.4. Client Options

(1) The `mount: volid port@server` option tells where to find the server that manages the volume. (Section 4.2, 4.4.)

(2) The `pagelock: SH or EX` option determines the default lock mode for pages. It replaces a global variable. (Section 4.2, A.1)

(3) The `locktimeout: # 10-second intervals` optin limits the amount of time the application will await a lock before the lock request fails with esmLOCKBUSY. (Section 4.2, A.3.)

5. Server

5.1. Server Options

(All the server options are described in Section 5.2.2.)

(1) The `wrapcount` option is seldom, if ever needed. the server figures out what wrap count it needs to use for the volumes listed in the server’s configuration files. (If volumes are added later, the server might have to be restarted with a higher wrap count.)

(2) The `background` option is no longer needed for users of csh(1) and other shells that have job control; the server figures out if it is in the background. (Bourne shell users still need this option.)

(3) The `maxvolumes` option is no longer available; the server figures out what to use by reading the configuration file.

(4) The `extrastats` option is no longer available; the stats are no longer printed.

(5) The `preflushlog` option is no longer available; it is always on.

(6) The `logformat, tempformat, dataformat` and `maxaddvolumes` options are new configuration-file options; they tell the server what volumes to manage. (Section 5.1, 5.2.2)

5.2. Server Commands

The server's terminal interface has been expanded to include these commands (described in section 5.2.3):

(1) `listdistr` prints two lists two sets of distributed transactions: the transactions for which this server is the coordinator, and the prepared transactions for which it is a subordinate server.

(2) `progress` toggles the `progress` option's value.

(3) `user` determines whether the server will print messages when a user (application) error is encountered.

(4) `source` reads a set of commands from a file.

(5) `redirect` redirects output to a file.

(6) `listmount` lists the volumes that are in use.

(7) `listvolumes` lists the volumes in the mountable-volume table.

(8) `addvolume` adds a volume to the mountable-volume table.

(9) `rmvolume` removes a volume from the mountable-volume table.

6. Formatting Volumes

Formatting volumes is done differently. The information regarding the format of a volume is now given to the formatter through configuration options, found in a configuration file. (Section 5.1.5)

7. Bugs Fixed

All known bugs from V2 have been fixed. Numerous efficiency improvements have made.