

---

# ***Versant Database Administration Manual***

**Release 7.0.1.4**

---



### **Versant History, Innovating for Excellence**

In 1988, Versant's visionaries began building solutions based on a highly scalable and distributed object-oriented architecture and a patented caching algorithm that proved to be prescient.

Versant's initial flagship product, the Versant Object Database Management System (ODBMS), was viewed by the industry as the one truly enterprise-scalable object database.

Leading telecommunications, financial services, defense and transportation companies have all depended on Versant to solve some of the most complex data management applications in the world. Applications such as fraud detection, risk analysis, simulation, yield management and real-time data collection and analysis have benefited from Versant's unique object-oriented architecture.

For more Information please visit [www.versant.com](http://www.versant.com)

### **Versant US**

Versant Corporation

255 Shoreline Drive, Suite 450, Redwood City, CA 94065

Ph +1 650-232-2400, Fx +1 650-232-2401

### **Versant Europe**

Versant GmbH

Wiesenkamp 22b, 22359 Hamburg, Germany

Ph +49.40.60990-0, Fx +49.40.60990-113

© 2008 Versant Corporation.

All products are trademarks or registered trademarks of their respective companies in the United States and other countries.

The information contained in this document is a summary only.

For more information about Versant Corporation and its products and services, please contact Versant Worldwide or European Headquarters.

# Table of Contents

<b>CHAPTER 1: Database Users and Authentication .....</b>	<b>13</b>
Versant Database Users .....	14
Type of Database Users.....	14
Database System Administrator - DBSA .....	14
Database Administrator - DBA.....	15
Database User - dbuser .....	16
Database User list .....	16
Root User — root (UNIX) .....	16
Managing Database Users.....	17
Utility Access Privileges for Database Users .....	18
User Authentication .....	20
Database Administrator Authentication .....	21
DBA Authentication for Specific Utilities.....	21
DBA Authentication for Utility APIs .....	22
DBA Authentication for Session Connection .....	22
DBA Authentication for Vedding(FTS).....	23
Changing the DBA Password.....	23
Database User Authentication.....	25
User Authentication Procedure .....	25
User Authentication using Third Party Plug-ins.....	26
LDAP Plug-In Example .....	27
Backward Compatibility .....	29
Customized User Authentication.....	29
User Authentication Program Structure.....	30
User Authentication Process .....	32
User Privileges .....	35
PUBLIC Access .....	35
Read-Write Access .....	35
User Authentication Sample Programs .....	35
For UNIX .....	35
Authentication program for application machine .....	36
Authentication program for database machine .....	37
Kerberos authentication program for database machine .....	38
Kerberos authentication program for application machine .....	41
For Windows .....	43

Sample client and server authentication programs for Windows .....	44
<b>CHAPTER 2: Database Creation .....</b>	<b>53</b>
Database Creation Basics .....	54
Prerequisites For Database Creation .....	54
Database Root Directory .....	54
Database Owner (DBA) .....	54
Database System Identifier File (osc-dbid) .....	55
Database Volumes .....	56
Usage Notes for UNIX Users .....	56
Parameters for Creating Database Directories .....	58
Mandatory Parameters .....	58
Database name .....	58
Database type .....	58
Database owner .....	58
Optional Parameters .....	58
Database size .....	59
Database volume file names .....	59
Locking .....	59
Logging .....	59
Database Creation Procedure .....	60
Step 1. Setting up the database directories and files .....	60
Automatic Setup .....	61
Manual Setup .....	63
Step 2. Editing the database profile (profile.be) .....	64
Step 3. Creating the database .....	67
System volume .....	68
Physical log volume .....	69
Logical log volume .....	69
.sharemem .....	69
Step 4. Moving a database .....	69
Moving just the data volumes .....	70
Moving all volumes .....	70
<b>CHAPTER 3: Database Profiles .....</b>	<b>71</b>
Application Process Parameters .....	72
Overview .....	72
Functional Parameters .....	74
alias .....	74

---

connect .....	75
genericObject.....	76
loose_schema_mapping .....	76
parallel_write .....	78
signal_block .....	79
Tuning Parameters .....	79
dbconnect_timeout.....	79
estimated_connections .....	80
estimated_objects .....	80
heap_size (application) .....	80
heap_size_increment (application) .....	81
max_objects.....	81
swap_threshold.....	82
Scenario 1:.....	83
Scenario 2:.....	83
Scenario 3:.....	83
Server Process Parameters .....	84
Overview .....	84
Server process profile location .....	84
On Windows .....	84
On UNIX .....	84
Server process profile parameters types.....	85
Database Creation Parameters.....	89
extent_size.....	89
llogvol.....	90
plogvol.....	91
sysvol .....	92
Large File Support .....	92
On Unix.....	92
On Windows .....	92
Functional Parameters .....	94
async_buffer_cleaner.....	95
async_logger.....	95
auto_addvol .....	95
auto_addvol_threshold.....	96
auto_addvol_aggression.....	97
Optimal Automatic Add Volume Settings.....	97
commit_delete.....	98
commit_flush.....	98
custom_be_plugins .....	98

datavol .....	99
locking.....	101
logging .....	101
Logging Usage Notes .....	101
startup_script .....	102
Monitoring startup executable.....	103
startup_script Usage Notes .....	103
Example.....	104
versant_be_dbalogginglevel .....	105
versant_be_dbalogginglevel .....	105
versant_be_logfile .....	105
be_syslog_level .....	105
User-Authentication using Third Party plug-ins.....	106
Tuning Parameters.....	106
bf_dirty_high_water_mark.....	108
bf_dirty_low_water_mark .....	108
class .....	109
db_timeout .....	109
heap_multi_region .....	109
heap_size (database) .....	110
heap_size_increment (database).....	110
index .....	111
logvolmaxsize .....	111
llog_buf_size .....	111
lock_batch_size .....	112
lock_wait_timeout .....	112
max_page_bufs .....	113
max_sorting_memory_per_query .....	113
max_sorting_memory_total .....	113
multi_latch.....	114
obe_port_begin .....	114
obe_port_end.....	114
plog_buf_size.....	115
polling_optimize .....	115
spin_count.....	116
transaction .....	116
user .....	116
volume .....	117
Size Limits Of Various Configurable Parameters.....	117
Statistics Parameters .....	119
assertion_level .....	119

---

stat .....	119
Connection or database statistic.....	120
Tracing Parameters.....	121
trace_comps .....	121
trace_entries .....	122
trace_file .....	122
Authentication Parameter.....	123
authentication_program .....	123
be_permit_pre70_clients.....	123
Event Notification Parameters.....	124
event_daemon .....	124
event_msg_mode .....	125
event_msg_transient_queue_size .....	125
event_registration_mode .....	125
event_daemon_notification .....	126
Database Space Parameter .....	126
out_of_space_warning_threshold .....	126
out_of_space_warning_threshold_increment .....	127
<b>CHAPTER 4: Database Utilities .....</b>	<b>129</b>
Quick Reference.....	130
Utility Reference .....	135
addvol.....	135
Usage Notes for Unix users .....	137
cleanbe.....	138
cleanfe.....	138
comparedb .....	138
compardb .....	138
convertdb .....	139
cnvrtdb .....	139
Conversion of Versant Database Release 6.x to Release 7.0 .....	142
createdb .....	142
createreplica.....	145
creatrep .....	145
db2tty .....	148
dbid .....	149
dbinfo .....	150
Restrictions and Suggestions for Read-only mode.....	152
dblist.....	154

dbtool .....	155
Get information about a Lock.....	156
Sample Output .....	157
Get information about an Object .....	160
Sample Output .....	160
Get information about Transactions .....	160
Get information about Database System .....	161
Sample Output .....	162
Get information about Storage Volumes .....	164
Cluster instances of a Class .....	165
Get information about Indexes: create and delete index .....	165
Create Index .....	167
Delete Index.....	167
Load Index .....	168
Viewing consistency and maintenance of AT Table .....	168
Trace components of a database .....	169
Get Event Notification information .....	173
Get information about all Classes in a database .....	173
Get information about Logging Activities .....	174
Determine fragmentation information and free space in database volumes.....	176
Get information about Schema Evolution .....	179
Explicitly evolve all instances to the latest schema.....	180
dbuser .....	182
dropclass.....	185
dropcls.....	185
dropinst .....	186
ftstool.....	188
Typical sequence of events .....	191
Restoring a replica database .....	191
makedb .....	193
makeprofile.....	196
NetworkServices .....	198
About the Versant system service .....	199
Adding additional ports and services .....	200
Add a service .....	201
Delete a service .....	202
Update a service .....	202
Restrictions on service name .....	202
Record the changes .....	202
oscp.....	203



---

polling.....	204
removedb .....	204
removereplca.....	206
removrep .....	206
reorgdb.....	207
Steps to reorganize a database.....	207
sch2db.....	207
Functionality .....	209
Generating schema files .....	209
Schema Evolution .....	210
Sch2db Usage notes .....	211
schcomp.....	211
Embedding arrays .....	215
Backward Compatibility .....	216
setdbid.....	217
ss.d.....	218
Usage notes.....	219
startdb .....	219
stopdb .....	220
vbackup.....	222
Command Parameters.....	224
(-backup) Backup database.....	224
Usage Notes .....	225
Backup database and start polling.....	225
(- restore) Restore database.....	226
Failure conditions in Restore .....	227
(- rename) Restore and Rename database .....	228
(- overwrite) Restore, rename and overwrite .....	229
Warm standby (Incremental Restore) .....	231
(-log) Start Roll Forward archiving .....	231
(-off) Stop Roll Forward archiving .....	232
(-info, -list) Backup Info .....	233
- info .....	233
- list .....	233
Options Parameters .....	235
(- level) Backup level .....	235
(- device) Backup device .....	236
(- position) Backup position .....	236
(- capacity) Backup Tape Capacity .....	237
(- blocking) Backup Blocking .....	238

(- comment) Backup comment .....	238
(- rollforward) Turn roll forward archiving ON.....	239
(- aggression) Aggression - Buffer time .....	239
(-script) Script .....	239
(-getbeprofile) Extract Server Process Profile File.....	241
(-odir) Target Directory for Extracted Server Process Profile File.....	242
Changing the parameter settings.....	242
Database Backup Frequency.....	243
Database Restoring Procedure.....	244
vcopydb.....	245
vinfo.....	248
verr .....	250
verrindx .....	251
vinstinf .....	251
vlicchk .....	252
vlicvrfy .....	253
vmovedb.....	253
vstats .....	255
Commands .....	255
Usage .....	261
Sample Output .....	261
Viewing Input Options .....	263
Viewing Output Options .....	264
Statistics File .....	265
Fault Tolerant Server option .....	265
Sample Output .....	266
vstream .....	269
Mode .....	269
Options .....	270
Specifying levels to export .....	273
Synchronizing schema .....	274
Operating across platforms and releases .....	275
Estimating operating system resources .....	275
Estimating stack usage .....	277
Estimating time .....	277
Comparison with vimport and vexport .....	278
vtape .....	278

---

<b>CHAPTER 5: Directories and Files .....</b>	<b>281</b>
Installed Directories and Files .....	282
Versant Root Directory Structure.....	282
Root and System Directories .....	285
Software root directory.....	285
Ant directory.....	286
bin directory (General executables).....	286
demo directory .....	287
doc directory .....	288
h directory (header files) .....	290
jre Directory .....	291
lib directory .....	291
uninstaller directory.....	292
Database Directories.....	293
Database root directory .....	293
Database Directory .....	294
Database Volumes and Files .....	294
System Volume.....	295
Physical log volume .....	295
Logical log volume .....	295
Application Process Profile File .....	296
LOGFILE.....	297
DBA Utility Logging.....	298
Server process profile.....	300
Lock file.....	300
Password file .....	301
Backup File .....	301
Shared Memory File .....	301
Database type file .....	302
Trace Log File.....	303
Configuration Files .....	303
Machine configuration file .....	304
System information file .....	304
Statistics File .....	305
Other Important Files.....	308
Database System Identifier File (osc-dbid) .....	308
Logical object identifier - LOID.....	308
License File (license.xml).....	312

<b>CHAPTER 6: Configuration Parameters .....</b>	<b>315</b>
Overview .....	316
Configuration Parameters for UNIX.....	318
UNIX Configuration Parameters Location .....	318
UNIX Configuration Parameter Considerations .....	320
UNIX Configuration Parameter Procedures .....	322
Environment variable .....	322
User configuration file .....	322
Machine configuration file .....	323
System information file .....	324
UNIX Configuration Parameters - Mandatory .....	324
PATH .....	324
UNIX Configuration Parameters - To Be Set In Special Cases .....	325
VERSANT_ROOT .....	325
VERSANT_DBID_NODE .....	326
VERSANT_DBID .....	327
VERSANT_DB .....	328
VERSANT_CFG .....	329
UNIX Configuration Parameters - For Asserting Values .....	329
VERSANT_REL, VERSANT_REL@node .....	329
VERSANT_ROOT@node .....	333
VERSANT_DBID_NODE@node .....	335
VERSANT_DBID@node .....	336
VERSANT_DB@node .....	337
UNIX Error Logging and Debugging Parameters .....	339
versant_fe_errlog .....	339
UNIX Statistics Collection Parameters .....	339
VERSANT_STAT_FILE .....	339
VERSANT_STAT_STATS .....	340
VERSANT_STAT_FUNCS .....	341
VERSANT_STAT_TIME .....	341
VERSANT_STAT_DBS .....	342
VERSANT_STAT_FLUSH .....	342
UNIX User Authentication Parameter .....	343
VERSANT_AUTH .....	343
UNIX Communications Parameter .....	343
Versant TCP/IP service name .....	343
VERSANT_SERVER_PORTS .....	346
UNIX Examples .....	347

---

UNIX Database Server System Parameter .....	350
Configuration Parameters for Windows.....	351
Windows Configuration Parameter Considerations .....	351
Windows Configuration Parameters - Mandatory .....	352
PATH .....	352
Windows Configuration Parameters - To be set in Special Cases .....	352
VERSANT_ROOT .....	353
VERSANT_DBID_NODE .....	353
VERSANT_DBID .....	354
VERSANT_DB .....	354
VERSANT_USER .....	355
VERSANT_HOST_NAME .....	355
Windows Configuration Parameters - Optional .....	356
TZ .....	356
HOSTNAME .....	356
ETC.....	356
Windows Configuration Parameters - For Asserting Values .....	357
VERSANT_REL, VERSANT_REL@node .....	357
VERSANT_ROOT@node.....	357
VERSANT_DBID_NODE@node .....	358
VERSANT_DBID@node.....	359
VERSANT_DB@node .....	359
Windows Error Logging and Debugging Parameters .....	360
Versant_fe_errlog .....	360
Windows Statistics Collection Parameters .....	361
VERSANT_STAT_FILE .....	361
VERSANT_STAT_STATS .....	362
VERSANT_STAT_FUNCS .....	362
VERSANT_STAT_TIME .....	363
VERSANT_STAT_DBS .....	363
VERSANT_STAT_FLUSH.....	364
Windows User Authentication Parameter .....	364
VERSANT_AUTH.....	364
Windows Parameters .....	364
VERSANT_SERVICE_NAME.....	364
VERSANT_SERVER_PORTS.....	367
Windows Examples .....	369

<b>CHAPTER 7: Error Codes.....</b>	<b>373</b>
Error Handles, Numbers and Messages .....	375
Virtual System Errors, Series - 0001 .....	376
System Level Errors, Series - 0700.....	384
Versant Server Errors, Series – 1000, 2000.....	388
Versant Network Errors, Series - 3000 .....	397
Versant Manager Errors, Series - 4000 .....	401
Locking Errors, Series - 5000 .....	407
Archive Errors, Series - 5200 .....	409
Query Errors, Series - 5400 .....	411
Version Errors, Series - 5600 .....	415
Schema Errors, Series - 6000 .....	416
Container Errors, Series - 6200.....	419
Event Notification Errors, Series - 6500 .....	421
Virtual Attribute Errors, Series - 6600.....	423
System Utility Errors, Series – 7000, 9000, 11000, 12300 .....	424
C++/Versant Errors, Series – 8000, 40000 .....	498
GUI Tool Errors, Series – 10000 .....	530
SQL & ST Errors, Series - 20000 .....	537
Versant Statistics Errors, Series - 30000 .....	544
 <b>CHAPTER 8: Microsoft Cluster Server Support.....</b>	 <b>563</b>
Support for Microsoft Cluster Server (MSCS) on Windows (Wolfpack).....	564
Versant MSCS Configuration Guide.....	564
Install Versant on nodes .....	564
Using Cluster Administrator .....	565
Viewing the Default Groups .....	565
Creating a New Versant Group .....	566
Generic Service.....	567
Host File Changes.....	569
Creating New osc-dbid File .....	569
Index.....	571

---

This Chapter gives details about the Database users, their privileges and their management. It also gives information about the Database user authentications and Customized user authentication.

Following topics are covered:

- Database Users
- Utility Access Privileges for All Users
- User Authentication
- Database Administrator Authentication
- Database User Authentication
- User Authentication Process
- User Privileges

## VERSANT DATABASE USERS

The Versant database can be accessed by different type of database users.

A database user is a role adopted by a person who creates, administers or accesses Versant databases, with special privileges.

Each user is identified by a unique user name that is stored in the database.

A user name can be of 32 characters in length.

## Type of Database Users

Versant has three kinds of database users. Their role differs depending on the type of the user.

They are:

- Database System Administrator (DBSA)
- Database Administrator (DBA)
- Database User (dbuser)

## Database System Administrator - DBSA

For UNIX installations, the database system administrator or "DBSA" is a user who as a superuser, and installs Versant on each machine in a network.

- There is only one DBSA for a system of Versant installations.
- The DBSA owns the `osc-dbid` file for a database system.
- The DBSA owns all Versant software root directories, including the `bin`, `h`, and `lib` subdirectories and all files under those directories, for all installations in a network. This ownership extends to all versions of Versant installed on a system of databases.
- The DBSA owns all Versant database root directories for all installations in a network.
- The DBSA owns all system information files for all installations in a network.



---

## Database Administrator - DBA

For all installations and operating systems, the database administrator or "DBA" of a database is the user who can create a database.

The DBA uses the `makedb` and `createdb` utilities to create the database.

**For more information, please refer to the “makedb” and “createdb” utilities.**

The primary function of the DBA is to create Versant databases, grant and administer user access to Versant databases and Backup Versant Databases.

- There can be only one DBA for any database.
- There can be any number of DBAs in a Versant system of distributed databases.
- The DBA should have an OS account. The DBA user name should correspond to OS user name.
- The DBA owns the database and related files and directories. After a database has been created, you can change its DBA by changing the ownership of its database directory.
- The DBA has access to the database even if the DBA name is not in the database user list.
- Only a DBA can delete the files in a database directory.

### Password Authentication for DBA

Versant introduces a password-based authentication for the DBA on top of the existing OS dependency.

The `makedb` utility will give the user an option to input a password to be associated with DBA. If the password is specified, it is stored in a new file `.pw` (For UNIX) and `pw` (For Windows) in the respective database directory.

If the user does not specify a password for the DBA with the `makedb` utility, only the OS check is performed. The DBA can also be associated with a password later, by using `dbuser` utility with “`-chpasswd`” option.

**For more information, please refer the “makedb” and “dbuser” utility.**

### Password Authentication for a Database

If incase the database is associated with a password, then some utilities will prompt the user to specify the DBA password before making any changes to the database.

User is allowed a maximum of three retries in case of a wrong password entry; after which an error will be returned.

**The utilities which need password authentication are:**

`createdb`, `addvol`, `comparedb`, `createreplica`, `dbinfo`, `dbtool`, `dbuser`, `ftstool`, `removedb`, `startdb`, `stopdb`, `vbackup`, `vcopydb`, `vmovedb`, `removereplia` and `setd-bid`.

## Database User - dbuser

The database user is a normal user, who has access to a particular database.

- The database user can act only according to the account privileges that he has been given by the DBSA or DBA.
- The database user may or may not have a valid operating system user account.
- A database user can change its password using the database user management utility (`dbuser`).

Versant maintains a list of logical users who have been given the access to a particular database by the DBSA or DBA.

## Database User list

For all installations and operating systems, Versant maintains a list of users who have access to a particular database, called the "database user list".

- The database user list, for a particular database is created and maintained by the DBA of that database.
- Each database has only one database user list. There may be a different database user list defined for each of the many databases in a network.
- The members of the database user list can connect to and use a particular database and can execute utilities to get information about a database system.

## Root User — root (UNIX)

For UNIX installations, a root user is the user with operating system root privileges for a particular machine in a network.

---

Being a root user with privileges deriving from operating system permissions provides only minimal privileges with respect to Versant. It is not required to use Versant as the root user.

**Example**

For example, on a UNIX system, if the database root directory is `/versant/db` and a database has been created using `/versant/db/db1` as a database directory, then the DBSA is the user that owns `/versant/db` and the DBA for database `db1` is the user who owns both `/versant/db/db1` and the database volumes that comprise the database.

**The database users access the database using applications and some utilities which are explained below:**

## Managing Database Users

The database users can be managed by the `dbuser` utility.

This user management utility (`dbuser`) is used to do many things related to the user. For example - to add, delete or to list the users of a database and to change password of a database users.

**For more information, please refer the “dbuser” utility.**

## Utility Access Privileges for Database Users

Following are the utility access privileges for each database user. Also shown are the utilities, which can be executed remotely.

These privileges are the same whether a system utility is executed from a command line or from the Versant Utility Tool.

Utility Name	dbsa	dba	dbuser	Remote Execution
addvol	no	yes	no	yes
comparedb	no	yes	no	yes
compardb (pc)	no	yes	no	yes
convertldb	no	yes	no	no
cnvrtdb (pc)	no	yes	no	no
createdb	no	yes	no	yes
createreplica	no	yes	no	yes
creatrep (pc)	no	yes	no	yes
db2tty	no	yes	yes	yes
dbid	yes	yes	yes	yes
dbinfo	no	yes	no	yes
dblist	yes	yes	yes	yes
dbtool	no	yes	no	no
dbuser	no	yes	no*	yes
dropclass	no	yes	yes	yes
dropcls (pc)	no	yes	yes	yes
dropinst	no	yes	yes	yes
ftstool	no	yes	yes	yes
makedb	no	yes	no	yes
makeprofile	no	yes	no	yes
makeprof (pc)	no	yes	no	yes
oscp	no	yes	yes	yes

---

polling	no	yes	yes	yes
rebilink	no	yes	yes	yes
removedb	no	yes	no	yes
removereplca	no	yes	no	yes
removrep (pc)	no	yes	no	yes
reorgdb	no	yes	no	no
sch2db	no	yes	yes	yes
setdbid	no	yes	yes	yes
ss.d	yes	yes	yes	yes
startdb	no	yes	no	yes
stopdb	no	yes	no	yes
vbackup	no	yes	no	yes
vcopydb	no	yes	no	yes
verr	no	yes	yes	no
verrindx	no	no	no	no
vmovedb	no	yes	no	yes
vstats	no	yes	yes	yes
vstream	no	yes	no	yes

**NOTE:-** \*A registered database user (dbuser) can change the password with dbuser -chpasswd.

## USER AUTHENTICATION

User Authentication is the means of identifying the database user in a unique way.

User identification is the basis of Versant's authorization mechanism. The users are allowed access to the system, only when identified as 'authorized' users by the system.

To prevent unauthorized use of a database, Versant provides user identification via different methods for the database users:

- Database Administrator Authentication
- Database User Authentication
- Customized User Authentication

## DATABASE ADMINISTRATOR AUTHENTICATION

As Database Administrators (DBAs) have valid Operating System account, Versant relies on Operating System mechanisms to verify the DBA's identity.

Versant introduces password-based authentication for the DBAs on top of the existing OS dependency.

The utility `makedb` will give the user an option to specify a password to be associated with DBA.

If a password is specified, it is stored in a new ".pw" file in the respective database directory. The password file will be created with READ/WRITE permission for the DBA only.

If the DBA does not specify a password with the `makedb` utility, no additional check other than OS check is performed for the DBA. A DBA can be associated with a password later using `dbuser` with "`-chpasswd`" option.

Password is a literal string and the length of the password can not exceed 255 characters and there should be no "space" character in the password.

The string "null" is reserved for null password. If the DBA does not associate with a password, he can specify the string "null" as password (or does not specify a password at all).

## DBA Authentication for Specific Utilities

Some of the utilities in Versant require DBA authentication.

The following utilities will prompt the user for password if the database is password protected.

<code>addvol</code>	<code>removedb</code>	<code>dbuser</code>
<code>comparedb</code>	<code>removereplica</code>	<code>ftstool</code>
<code>createdb</code>	<code>setdbid</code>	<code>vcopydb</code>
<code>createreplica</code>	<code>startdb</code>	<code>vmovedb</code>
<code>dbinfo</code>	<code>stopdb</code>	
<code>dbtool</code>	<code>vbackup</code>	

**NOTE:-** If the DBA enters a wrong password, he gets only three chances to re-enter the password.

## DBA Authentication for Utility APIs

For some utility APIs, you can specify the password through `-password` option.

These utility APIs are: `o_makedb`, `o_createdb` etc.

Here the caller can specify DBA's password through "`-password`" option if DBA is password protected.

For the DBA utility API `o_setdbid`, the caller specifies DBA's name and password through `o_setuserlogin` API.

### Examples illustrating the use of utility APIs with passwords

Following are examples illustrating usage of utility APIs with passwords:

#### DBA Utility APIs:

##### 1. Using `o_createdb`:

```
o_nvlist nvlist;
o_createnvlist(&nvlist);
o_addtonvlist(nvlist, "-password", "mypassword");
o_createdb(dbname, nvlist);
```

##### 2. Using `o_setdbid` with `o_setuserlogin`:

```
o_userInfo userinfo;
strcpy(userinfo.username, "DBA's name");
strcpy(userinfo.password, "DBA's password");
o_setuserlogin(&userinfo);
o_setdbid(dbid, dbname);
```

The utility APIs will not use the information set by the `o_setuserlogin` API, but `o_setdbid` is an exception. The `o_setdbid` API gets the information of user set by the `o_setuserlogin` API. Regardless of whether the thread is in a session or not, the user must provide the utility API with the DBA password if the database is password protected.

## DBA Authentication for Session Connection

Applications that connect to the database remotely should provide a password for the user connecting to the database. This is also applicable to DBA connecting to the database



remotely. The application should provide the password using the `o_setuserlogin` API as described in the C/Versant Reference Manual. If the password is not specified during connection, the authentication will fail with an error if database is password protected.

**NOTE:-** In Versant, a session connection to the server implicitly starts the server even if the user calling the session connection is not the DBA. After the database has started implicitly, if the user performs any database administration tasks, the password should be supplied as part of that API for verification.

## DBA Authentication for Vedding(FTS)

For FTS, the DBA must ensure that the same password is used for the replica pair.

If the passwords are different, the utility `createreplica` will return an error while connecting to the database.

## Changing the DBA Password

The `dbuser` utility can be used to change the password of the DBA.

(The user must be logged into the OS as the DBA to run this command.)

```
dbuser -chpasswd -n <dbaname> -passwd <newpassword> -opasswd  
                                     <oldpassword>
```

### Failed Authentication Errors

If the password verification fails, `SM_E_INVALID_PASSWORD` error is returned.

If the user is not logged into the OS as the DBA, while changing the password for DBA, `UT_ER_USER_NOTDBA` error is returned.

### Password Recovery

If the DBA forgets the password, the DBA can login as an OS user into the machine and delete the password file.

Once the password file is deleted, the DBA authentication will bypass the password check and the DBA can gain access to the database.

In such a case the DBA will have to create a new password using the `dbuser` utility to protect against any remote DBA activities.

## DATABASE USER AUTHENTICATION

Versant authenticates the users attempting to connect to a database by using the information stored in that database.

Versant implements password-based authentication for users with a valid password and Operating System based authentication for users with no password. During Operating System based authentication, no passwords are checked. The requirement is that the user be a valid OS user.

The DBA can choose to add a user with no password, in which case the authentication mechanism at the server-side for that user is Operating System based. For users with a password, password based authentication is followed at the server-side.

The client is authenticated at the server during the initial connect (`o_beginsession` or `o_connectdb`). Once the connection has been established, no further authentication is done during the life of that connection.

The client has a secret password, known only to the client and the server. The password cannot exceed 255 characters. The client sends this password to the server to prove its identity. The server verifies the client's password with the information stored in the database to authenticate the client.

## User Authentication Procedure

To protect password confidentiality, Versant encrypts the password before sending it across the network.

The following API is used by the client, to set the user information required for user authentication.

The user information is specific for the caller thread of this API.

The user information is sent to the server during a connect (`o_beginsession()` or `o_connectdb()`).

```
o_err o_setuserlogin(o_userInfo userinfo /*
user information */);
struct o_userInfo {char user name [O_NAME_SIZE];
char password [O_PASSWD_SIZE];
};
```

If the user information is not set using this API before connecting to a database, the user name from the operating system will be picked up as the default.

In case the user entry in the database has no password, the server assumes Operating System authentication and performs the corresponding check for that user during the initial connect. When performing Operating System level authentication, it is required that the user exists as a valid Operating System user, both on the client and the server side. Failing to satisfy both the conditions would result in an authentication failure. For the case where, the user entry in the database has a valid password, only password-based authentication is performed. Failing to provide the correct password would result in an authentication failure.

**The examples below describe the authentication procedure.**

- `dbuser -add -n user1 -passwd "" <dbname>` (A user with no password)
- `dbuser -add -n user2 -passwd user2-password <dbname>` (A user with a password)
- `db2tty -d <dbname>` (If the user invoking the `db2tty` process is same as the user1, this command will be authenticated based on Operating System)
- `db2tty -d <dbname>` (If the user invoking the `db2tty` process is same as the user2, this command will fail as it performs password based authentication)
- `dbuser -ch passwd -n user1 -passwd user1-password` (The user invoking this command must be user1, for it to succeed)
- `db2tty -d <dbname> -u user1 -p user1-password` (Will succeed from any machine and as any OS user)

## User Authentication using Third Party Plug-ins

Versant provides a plug-in mechanism of authenticating the user's password using plug-ins like LDAP.

The DBA must provide the system with the location of the plug-in using the backend profile parameter "`custom_be_plugins`".

**For more information, please refer to the "`custom_be_plugins`" parameter, in the Chapter "Database Profiles" in the *Versant Database Administration Manual*.**

The basic idea is that, if a plug-in for user authentication is installed, instead of using Versant internal authentication mechanism, Versant database server will use mechanism provided by the plug-in to authenticate the user.

The plug-in is under the form of a DLL. The DLL will be loaded when the Versant database server starts and the initialization routine in the DLL will be called to install the user provided

---

authentication routine. An example is given by the following to illustrate how to implement a plug-in using a LDAP as authentication system.

**NOTE:-** User Authentication using Third Party Plug-ins will override the password based authentication. The access right of the plug-in file should be at least "read" for the DBA.

DBA cannot use "dbuser" to change user's password. One should use the framework provided by the third party authentication mechanism.

The third party authentication mechanism will not be used to authenticate pre-7.0 clients.

## LDAP Plug-In Example

**Step 1:** Write the plug-in initialization function for the plug-in.

The purpose of this function is to tell Versant database server which function in the DLL should be called to authenticate a user. This function is simple:

1. It takes a "o\_ptr" as input argument;
2. In the function, the user just needs to initialize a "o\_vcpuserdesc" structure with the pointer of the "function" (in the example, the function "userPwVerification") which really does user authentication. Then call MACRO "VCP\_USERINSTALL" to install the authentication plug-in. The first argument of the MACRO is the name of the plug-in. It should be always "versant-user-info". The second argument of the MACRO is the "o\_vcpuserdesc" structure which gives the pointer of the authentication function.
3. The name of plug-in's initialization function should be in accordance with the name of the DLL. Suppose in our example, the name of the DLL which will contains our plug-in is "libuserpw.so". The name of the initialization function should be given by observing the following rules:
  - Strip the leading "lib" or "dll" string from the name of the DLL (that gives string "userpw.so" in our example);
  - Strip the ".so" or ".sl" or ".dll" suffix from the remaining string (that gives string "userpw" in our example);
  - Capitalize the first character of the remaining string and put the rest on lower case (that gives "Userpw" in our case);
  - Finally, append "\_Init" (that gives "Userpw\_Init" in our case).

The resulting string will be the name of the initialization function in the DLL. This function will be called by the database server as soon as the DLL is loaded in the memory.

```
#include "osc.h"
/* plug-in initialization function "Userpw_Init"
 * This function will be called right after the DLL
 * is loaded in memory. And the name of the function
 * should be in accordance with the name of the DLL that
 * contains these functions. */

int Userpw_Init(o_ptr vcparg)

/* structure which will be
 * used to install the plug-in */

ovcpuserdesc vuser;
int error = O_OK; /* for error handling */

/* fill the structure with the pointer of the
 * authentication function */

vuser.user_verification = (vcpUserVerification)userPwVerification;

/* call VCP_USERINSTALL to install the
 * authentication function
 * Note that the name of the plug-in should called
 * "versant-user-info"
 */
error = (VCP_USERINSTALL("versant-user-info", &vuser));
return error;
```

### **Step 2: Write the authentication function.**

This function will be called to authenticate the user. A pointer pointing to a "o\_userInfo" structure is passed to this function. The "o\_userInfo" structure contains the name and password of the user to be authenticated. This routine will do whatever necessary to authenticate the user. It should return 0 if the user has been successfully authenticated and a non-zero value otherwise. In the following example, it is supposed to delegate the user authentication task to a LDAP server.

```
o_err userPwVerification(o_userInfo *userinfo)
char *username = userinfo->username;
char *password = userinfo->password;
ld = ldap_open(host, port);
if (ld == NULL)
```

---

```

return -1;
sprintf(dn, "cn=%s", username);
id = ldap_compare_s(ld, dn, "pw", password) ;
msgtype = ldap_result(ld, id, 1, 0, &res);
return ld->ld_error;

```

**Step 3:** Create the library and call it “libuserpw.so” with the above functions defined in it. Note, the “name” of the DLL is important (see step 1).

**Step 4:** Insert this line in the profile.be of the database:

```
custom_be_plugins <Path the directory where libuserpw.so resides>/libuserpw.so
```

## Backward Compatibility

As a default behavior, pre7.0 clients ( session connections and utility connections) will not be able to connect to 7.0 server.

The DBA can enable to a certain extent the pre7.0 clients to connect by using the backend profile parameter `be_permit_pre70_clients` (on/off).

Note that even if this parameter is set, pre 7.0-DBA-specific utility connections will not be permitted.

## Customized User Authentication

To customize user authentication, do the following.

### Create authentication programs.

Create two authentication programs: an application authentication program that will run on the machine containing the application, and a database authentication program that will run on the machine containing the database.

These programs will communicate with each other through channels established by Versant.

These programs can perform any actions that you want in order to come to a decision as to whether a particular user is valid.

**NOTE:-** Sample programs are provided in a later section of this chapter.

**Tell Versant about the authentication programs.**

On the application machine, set the environment variable `VERSANT_AUTH` to point to the application authentication program.

The general format for the environment variable is: `VERSANT_AUTH=program_path`

On the database machine, in the database profile file, add a line containing the keyword `authentication_program` followed by the path to the database authentication program.

The general format for the entry is: `authentication_program program_path`

## **Use two process model.**

Run your application in two processes by linking with the Versant two process system library for your platform: `liboscfe.a`, `liboscfe.lib`, or `liboscfe.so`.

After these steps, Versant will run the authentication programs everytime a database connection request is made, either with an interface routine or with a database utility.

## **User Authentication Program Structure**

While the authentication programs are running, the programs on the application and database machines can communicate with each other through a connection established by Versant.

The authentication programs can send any of the following operation codes to the Versant process to which they are connected:

### **`O_AUTH_READ`**

"Send to me one data packet you receive from your database connection."

### **`O_AUTH_WRITE`**

"Write my data packet to your database connection."

### **`O_AUTH_GRANTED`**

"This user is ok."

### **`O_AUTH_DENIED`**

"This user is not ok."

### **`O_AUTH_SET_TIMEOUT`**

"Set the timeout period to a specified number of seconds."



---

The default timeout is 600 seconds.

The structure of a packet that can be sent with these operation codes is:

```
typedef struct o_auth_packet {
    o_u2b      opcode;      /* operation type */
    o_u4b      len;         /* length of data in byte */
    o_ptr      data;        /* packet data */
} o_auth_packet;
```

To an authentication process, the communication pipe appears to be open file descriptor 0 and 1.

The authentication processes reads a packet from file 0 and writes a packet to file 1.

While reading a packet, the opcode and len fields of the packet are first retrieved from the connection and then data of the packet up to len bytes is read.

In the simplest case, an authentication program on an application machine should do the following to prepare an authentication program on the database machine:

1. Send O\_AUTH\_WRITE packet.
2. Send O\_AUTH\_READ packet.
3. Read packet from the communication pipe.
4. Process the incoming packet.
5. Send O\_AUTH\_GRANT or O\_AUTH\_DENIED packet.

The authentication program on the database machine should do the following:

1. Send O\_AUTH\_SET\_TIMEOUT.
2. Send O\_AUTH\_READ.
3. Read a packet from the communication pipe.
4. Process the authentication.
5. To grant authorization, send O\_AUTH\_GRANTED.
6. To deny authorization, send O\_AUTH\_DENIED.

The database administrator for a database must still add and delete users with the Versant dbuser utility.

The authentication programs are responsible for validating that a user is valid, but Versant will only extend to that user the privileges established by the database administrator.

## User Authentication Process

Versant user authentication process is as follows:

1. TCP/IP `inetd` daemon starts.

To use Versant, a machine must have TCP/IP software installed. When a machine with TCP/IP starts, a TCP/IP `inetd` daemon process is started that will listen for service requests.

2. A user requests a Versant session.

A user logs into the operating system, starts a Versant application, and makes a request to start a session (which requires a database connection.)

3. An authentication program is started on the application machine.

When it receives a "begin session" or "connect database" request, Versant checks to see if the `VERSANT_AUTH` environment variable has been set on its machine.

If the `VERSANT_AUTH` environment variable has been set, Versant reads `VERSANT_AUTH`, looks up the specified authentication program, forks a process, starts the authentication program, and redirects `stdin` and `stdout` so that they serve as a channel of communication between the application process and the authentication program.

The Versant application may be using Versant files on a local or remote machine. The check is made on the machine containing the `VERSANT_ROOT` location parameter in use by the application.

4. The application machine authentication program sends "`O_AUTH_WRITE`".

When it starts, the application authentication program sends an `O_AUTH_WRITE` packet containing a password to the application process. This message means, "please forward this password to the authentication program on the machine containing the database." (The username is sent by the Versant connection request.)

5. The application machine authentication program then sends "`O_AUTH_READ`".

After sending a username, the application authentication program sends an `O_AUTH_READ` packet to the application process. This message means, "when you get a message from the Versant server process, forward it to me."

6. The application sends a connection request and username to the database machine.

The request for a database connection and a password are sent to the `inetd` daemon process on the machine containing the database. (The same procedure is followed whether the database machine is remote or local.)

The `inetd` daemon process will read its TCP/IP configuration files and respond to the connection request by starting a Versant `ss.d` (system services) process and then connect the application with the `ss.d` process.

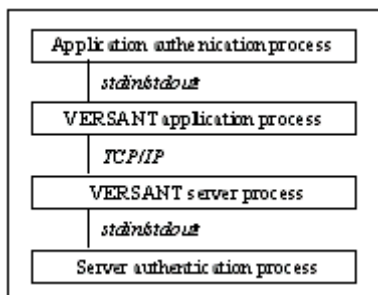
The `ss.d` process will start the database, if it is not already running, and then fork a database server process.

Once the connection request has been sent, control of the application process passes to the application authentication program, and the only action that the application process can perform is to pass messages to and from the authentication program.

7. An authentication program is started on the database machine.

The database server process will check its server process profile file. If it sees a line beginning with the keyword `authentication_program`, it forks a process, starts the program specified, and redirects `stdin` and `stdout` so that they serve as a channel of communication between the server process and the server authentication program.

At this point, the lines of communication are:



The authentication programs on the application and database machines are now active and can communicate with each other through the connection established by Versant. The authentication programs can send any of the following operation codes to the Versant process to which they are connected: `O_AUTH_READ`, `O_AUTH_WRITE`, `O_AUTH_SET_TIMEOUT`, `O_AUTH_GRANTED`, and `O_AUTH_DENIED`. (This discussion assumes the default timeout. To change it, you would send `O_AUTH_SET_TIMEOUT`.)

8. The database machine authentication program sends "`O_AUTH_READ`". When it starts, the database authentication program sends the packet operation code `O_AUTH_READ` to the database server process. This message means, "when you get a message from the Versant application process, forward it to me."

At this point, control of the database server process has passed to the database authentication program, and the only action that the server process can perform is to pass messages to and from the authentication program.

9. The authentication programs evaluate the username.

The authentication programs now conduct any kind of dialogue that you desire, communicating with each other through Versant. In this dialogue, each program can send a packet with `O_AUTH_WRITE` and then stand by to receive a packet with `O_AUTH_READ`.

For example, to send a message to the application authentication program, the database authentication program sends, on stdout, an `O_AUTH_WRITE` packet to the Versant server process. The Versant server process then forwards the packet to the Versant application process. If the application process has received an `O_AUTH_READ` packet, it will give the packet to the application authentication program.

### 10. If authorization is granted...

If the server authentication program decides that the user is okay, it sends the message `O_AUTH_GRANTED` to the Versant server process and then terminates.

When it receives `O_AUTH_GRANTED`, the Versant server process forwards the message to the Versant application process and returns. Its first action is to check whether the user is on its own list of authorized users. If the user is on the list, then it stays alive in order to service database requests from the application. If the user is not on its list, it sends an error to the application, closes the connection, and terminates.

When it receives `O_AUTH_GRANTED`, the Versant application process forwards the message to the application authentication program and returns.

When it receives `O_AUTH_GRANTED`, the application authentication program terminates.

If authorization is denied...

If the server authentication program decides that the user is not okay, it sends the message `O_AUTH_DENIED` to the Versant server process and then terminates.

When it receives `O_AUTH_DENIED`, the Versant server process forwards an error message to the

Versant application process and terminates.

When it receives `O_AUTH_DENIED`, the Versant application process forwards the error message to the authentication program, returns, and either handles the error or terminates with an error.

When it receives `O_AUTH_DENIED`, the application authentication program terminates.

---

## User Privileges

A privilege of a user, is the right to access a database or objects in that database, or execute database utilities.

### PUBLIC Access

When the PUBLIC access is granted for a database, anyone can access the database. No user authentication is done.

When this is true, the database is completely accessible to anyone. None of the above requirements need to be satisfied in order to gain access entry into that database system.

However it should be noted that setting the database to public access, does not guarantee entry into the database system while running in single-process mode. As stated earlier, only the DBA can access the database in single-process mode.

### Read-Write Access

- Read-Only
- Read Write

These access privileges can be granted to a database user by the DBA while adding a user to the system. The DBA always has read-write access privilege.

## User Authentication Sample Programs

### For UNIX

Following are sample programs for customized user authentication suitable for Unix. The communication mechanism between the authentication program and the application differs on Windows.

On Unix, the application communicates with the authentication program through a Unix pipe, which is duplicated onto the standard input and standard output handles of the authentication program. The program simply used standard i/o routines to read and write from the pipe.

## Authentication program for application machine

Following is a sample user authentication program to run on the machine running an application.

It is a stripped down example that simply passes a plain-text password to an authentication program running on the machine containing a database.

```
#include <stdio.h>
#include <stdlib.h>
#include <syslog.h>
#include <omapi.h>
/*
 * File descriptor 0 is opened for reading from Versant
 * kernel.
 * File descriptor 1 is opened for writing to Versant kernel.
 */

#define IN      0
#define OUT     1
#define PASSWORD "simple-but-unsafe"
#define SYSLOG(msg) syslog(LOG_ERR, "(FE): %s", (msg))
int main(
    int    argc,
    char   *argv[]
)
{
    o_auth_packet packet;
    SYSLOG("Start");
    /* write an O_AUTH_WRITE packet that contains the
 * password */
    packet.opcode = O_AUTH_WRITE;
    packet.len = strlen(PASSWORD);
    write(OUT, &packet, sizeof(o_auth_packet));
    write(OUT, PASSWORD, packet.len);
    /* write an O_AUTH_READ packet to server and wait
 * for an reply */
    packet.opcode = O_AUTH_READ;
    packet.len = 0;
    write(OUT, &packet, sizeof(o_auth_packet));
    read(IN, &packet, sizeof(o_auth_packet));
```

---

```

/* verify the reply packet */
switch (packet.opcode)
{
    case O_AUTH_GRANTED:
        /* forward the packet to kernel */
        SYSLOG("Access granted");
        write(OUT, &packet, sizeof(o_auth_packet));
        break;
    case O_AUTH_DENIED:
        /* forward the packet to kernel */
        SYSLOG("Access denied");
        write(OUT, &packet, sizeof(o_auth_packet));
        break;
    default:
        SYSLOG("Wrong reply packet");
        exit(2);
}
/* done */
SYSLOG("Done");
return 0;
}

```

## Authentication program for database machine

Following is a sample user authentication program to run on the machine containing a database.

It is a stripped down example that simply reads a plain-text password and then verifies it.

```

#include <stdio.h>
#include <stdlib.h>
#include <syslog.h>
#include <omapi.h>
/*
 * File descriptor 0 is opened for reading from Versant
 * kernel.
 * File descriptor 1 is opened for writing to Versant
 * kernel.
 */
#define IN 0
#define OUT 1
#define PASSWORD "simple-but-unsafe"

```

```
#define SYSLOG(msg) syslog(LOG_ERR, "(BE): %s", (msg))
int main(
    int    argc,
    char   *argv[]
)
{
    o_auth_packet  packet;
    char           buff[30];
    SYSLOG("Start");
    /* write an O_AUTH_READ packet to kernel and wait
 * for an reply */
    packet.opcode = O_AUTH_READ;
    packet.len = 0;
    write(OUT, &packet, sizeof(o_auth_packet));
    /* read in the password */
    read(IN, &packet, sizeof(o_auth_packet));
    read(IN, buff, packet.len);
    /* verify the reply packet */
    if (packet.len == strlen(PASSWORD) &&
        !strcmp(buff, PASSWORD, strlen(PASSWORD)))
    {
        SYSLOG("Access granted");
        packet.opcode = O_AUTH_GRANTED;
    }
    else
    {
        SYSLOG("Access denied");
        packet.opcode = O_AUTH_DENIED;
    }
    /* write the authentication result to kernel */
    packet.len = 0;
    write(OUT, &packet, sizeof(o_auth_packet));
    /* done */
    SYSLOG("Done");
    return 0;
}
```

### Kerberos authentication program for database machine

Following is a sample user authentication program to run on the machine containing a database.



It uses Kerberos V5 as the principal authentication mechanism. It performs the by parsing a KRB\_AP\_REQ packet sent from authentication program running on the application machine.

Many of the resource deallocation codes are ignored on purpose in order to make the sample program simpler.

```
#include "krb5.h"
#include "com_err.h"
#include <stdio.h>
#include <ctype.h>
#include <syslog.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <signal.h>
#include <omapi.h>
#include "auth_krb5.h"
#define SYSLOG(msg) syslog(LOG_ERR, "(BE): %s", (msg))
#define KEYTAB "/tmp/v5srvtab"
void main(argc, argv)
int argc;
char *argv[];
{
    o_auth_packet      packet;
    char*              msgp;
    krb5_context        context;
    krb5_auth_context* auth_context = NULL;
    krb5_ccache          ccdef;
    krb5_data            recv_data;
    krb5_principal       principal;
    krb5_keytab           keytab;
    krb5_error_code      retval;
    SYSLOG("Start");
    /* initialize Kerberos context */
    krb5_init_context(& context);
    krb5_init_ets(context);
    /* initialize authentication context */
    if (retval = krb5_auth_con_init(context, &auth_context))
    {
        SYSLOG(error_message(retval));
    }
}
```

```
        exit(1);
    }
    /* identify our credentials cache */
    if (retval = krb5_cc_default(context, &ccdef))
    {
        SYSLOG(error_message(retval));
        exit(2);
    }
    /* write an O_AUTH_READ packet to kernel and wait for an reply */
    packet.opcode = O_AUTH_READ;
    packet.len = 0;
    write(OUT, &packet, sizeof(o_auth_packet));
    read(IN, &packet, sizeof(o_auth_packet));
    if ((msgp = (char *) malloc(packet.len)) == NULL)
    {
        SYSLOG("Out of memory");
        exit(3);
    }
    /* read in the KRB_AP_REQ message */
    read(IN, msgp, packet.len);
    /* reconstruct the KRB_AP_REQ message */
    memcpy(&recv_data, msgp, sizeof(recv_data));
    recv_data.data = msgp + sizeof(recv_data);
    if (retval = krb5_kt_resolve(context, KEYTAB, &keytab))
    {
        SYSLOG(error_message(retval));
        exit(4);
    }
    /* parse the KRB_AP_REQ message */
    if (retval = krb5_sname_to_principal(
        context, OBE_HOST, OBE_SERVICE,
        KRB5_NT_SRV_HST, &principal))
    {
        SYSLOG(error_message(retval));
        exit(5);
    }
    if (retval = krb5_rd_req(context, &auth_context,
        &recv_data, principal,
        keytab, NULL, NULL))
    {
```

---

```

        SYSLOG(error_message(retval));
        SYSLOG("Authentication failed");
        packet.opcode = O_AUTH_DENIED;
    }
    else
    {
        SYSLOG("Authentication succeeded");
        packet.opcode = O_AUTH_GRANTED;
    }
    /* write the authentication result to kernel */
    packet.len = 0;
    write(OUT, &packet, sizeof(o_auth_packet));
    /* done */
    SYSLOG("Done");
    exit(0);
}

```

## Kerberos authentication program for application machine

Following is a sample user authentication program to run on the machine containing an application.

This program utilizes Kerberos V5 as the principal authentication mechanism. It achieves authentication by sending a `KRB_AP_REQ` packet to authentication program on a machine containing a database.

Many of the resource deallocation codes are ignored in order to make the sample program simpler.

```

#include "krb5.h"
#include "com_err.h"
#include <stdio.h>
#include <ctype.h>
#include <syslog.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <signal.h>
#include <omapi.h>
#include "auth_krb5.h"

```

```
#define SYSLOG(msg)syslog(LOG_ERR, "(FE): %s", (msg))
void
main(argc, argv)
int argc;
char *argv[];
{
    o_auth_packet      packet;
    krb5_context        context;
    krb5_auth_context* auth_context = NULL;
    krb5_ccache          ccdef;
    krb5_data            recv_data;
    krb5_error_code      retval;
    SYSLOG("Start");
    /* initialize Kerberos context */
    krb5_init_context(& context);
    krb5_init_ets(context);
    /* initialize authentication context */
    if (retval = krb5_auth_con_init(context, &auth_context))
    {
        SYSLOG(error_message(retval));
        exit(1);
    }
    /* identify our credentials cache */
    if (retval = krb5_cc_default(context, &ccdef))
    {
        SYSLOG(error_message(retval));
        exit(2);
    }
    /* format a KRB_AP_REQ message */
    if (retval = krb5_mk_req(context, &auth_context,
        AP_OPTS_USE_SESSION_KEY,
        OBE_SERVICE, OBE_HOST, NULL, ccdef, &recv_data))
    {
        SYSLOG(error_message(retval));
        exit(3);
    }
    /* send the KRB_AP_REQ message to kernel via an O_AUTH_WRITE
       packet */
    packet.opcode = O_AUTH_WRITE;
    packet.len = sizeof(recv_data) + recv_data.length;
```

---

```

if ((packet.data = (o_ptr) malloc(packet.len)) == NULL)
{
    SYSLOG("Out of memory");
    exit(4);
}
memcpy(packet.data, &recv_data, sizeof(recv_data));
memcpy(((char *)packet.data)+sizeof(recv_data),
        recv_data.data, recv_data.length);
write(OUT, &packet, sizeof(o_auth_packet));
write(OUT, packet.data, packet.len);
/* write an O_AUTH_READ packet to kernel and wait for an reply */
packet.opcode = O_AUTH_READ;
packet.len = 0;
write(OUT, &packet, sizeof(o_auth_packet));
/* wait */
read(IN, &packet, sizeof(o_auth_packet));
/* verify the reply packet */
switch (packet.opcode)
{
    case O_AUTH_GRANTED:
        /* forward the packet to kernel */
        SYSLOG("Authentication succeeded");
        write(OUT, &packet, sizeof(o_auth_packet));
        break;
    case O_AUTH_DENIED:
        /* forward the packet to kernel */
        SYSLOG("Authentication failed");
        write(OUT, &packet, sizeof(o_auth_packet));
        break;
    default:
        SYSLOG("Wrong reply packet received");
        exit(2);
}
/* done */
SYSLOG("Done");
exit(0);
}

```

## For Windows

Following are sample programs for customized user authentication suitable for Windows.

On Unix, the application communicates with the authentication program through a Unix pipe, which is duplicated onto the standard input and standard output handles of the authentication program. The program simply used standard i/o routines to read and write from the pipe.

On Windows we use an Windows anonymous pipe to communicate between the authentication program and the application. The read handle and the write handle of the pipe are passed in to the authentication program as command-line arguments. These arguments are integers in decimal notation, and need to be converted to handle numbers using `atoi()`. The authentication program is expected to use `WIN32 ReadFile()` and `WriteFile()` calls to read from and write to these handles in order to communicate with the spawning program.

The server authentication program follows the same protocol to communicate with Versant's server process. It takes input and output handle numbers as command-line arguments, and uses `ReadFile()` and `WriteFile()` for i/o on these handles.

## Sample client and server authentication programs for Windows

```
/*
 * A sample client user authentication program.
 */
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <omapi.h>
FILE *outputFile = NULL;
#define SYSLOG(msg) { \
    outputFile = fopen("authfe.out", "a"); \
    fprintf(outputFile, "(FE): %s\n", (msg)); \
    fclose(outputFile); \
}
#define PASSWORD "simple-but-unsafe"
/*
 * This is a bare-bone front-end authentication program
 * whose sole purpose is to demonstrate the usage of
 * Versant user authentication facility. It simply
 * passes a plain-text password to the server
 * authentication program.
 */
int main (int argc, char **argv)
{
    o_auth_packet    packet;
```

---

```

FILE          *pfile;
DWORD BytesWritten, BytesRead;
HANDLE hIn, hOut;
char message[256];
if (argc < 3)
    SYSLOG("Bad arguments");
hIn = (HANDLE)atoi (argv[1]);
hOut = (HANDLE)atoi (argv[2]);
sprintf (message, "hIn = %d, hOut = %d", hIn, hOut);
SYSLOG (message);
SYSLOG("Start");
/* write an O_AUTH_WRITE packet that contains the
 * password to kernel
 */
packet.opcode = O_AUTH_WRITE;
packet.len = strlen(PASSWORD);
if (WriteFile ( hOut, &packet, sizeof(o_auth_packet),
               &BytesWritten, NULL) == 0)
{
    sprintf (message, "WriteFile failed : %d",
             GetLastError ());
    SYSLOG (message);
    return (1);
}
if (BytesWritten != sizeof(o_auth_packet))
{
    sprintf ( message, "WriteFile wrote %d bytes
                     instead of %d", BytesWritten,
                     sizeof(o_auth_packet));
    SYSLOG (message);
    return (1);
}
if (WriteFile (hOut, PASSWORD, packet.len,
               &BytesWritten, NULL) == 0)
{
    sprintf (message, "WriteFile failed : %d",
             GetLastError ());
    SYSLOG (message);
    return (1);
}
if (BytesWritten != packet.len)

```

```
{
    sprintf (message, "WriteFile wrote %d bytes
instead of %d", BytesWritten, packet.len);
    SYSLOG (message);
    return (1);
}
/* write an O_AUTH_READ packet to kernel and wait for an reply */
packet.opcode = O_AUTH_READ;
packet.len = 0;
if (WriteFile (hOut, &packet, sizeof(o_auth_packet),
               &BytesWritten, NULL) == 0)
{
    sprintf (message, "WriteFile failed : %d",
            GetLastError ());
    SYSLOG (message);
    return (1);
}
if (BytesWritten != sizeof(o_auth_packet))
{
    sprintf (message, "WriteFile wrote %d bytes
instead of %d", BytesWritten, sizeof(o_auth_packet));
    SYSLOG (message);
    return (1);
}
if (ReadFile (hIn, &packet, sizeof (o_auth_packet),
             &BytesRead, NULL) == 0)
{
    sprintf (message, "ReadFile failed : %d",
            GetLastError ());
    SYSLOG (message);
    return (1);
}
if (BytesRead != sizeof (o_auth_packet))
{
    sprintf (message, "ReadFile read %d bytes
instead of %d",
            BytesRead, sizeof (o_auth_packet));
    SYSLOG (message);
    return (1);
}
```



---

```

/* verify the reply packet */
switch (packet.opcode)
{
case O_AUTH_GRANTED:
    /* forward the packet to kernel */
    SYSLOG("Access granted");
    if (WriteFile (hOut, &packet, sizeof (o_auth_packet),
                  &BytesWritten, NULL) == 0)
    {
        sprintf (message, "WriteFile failed : %d",
                  GetLastError ());
        SYSLOG (message);
        return (1);
    }
    if (BytesWritten != sizeof (o_auth_packet))
    {
        sprintf (message, "WriteFile wrote %d bytes instead of
                        %d", BytesWritten, sizeof(o_auth_packet));
        SYSLOG (message);
        return (1);
    }
    break;
case O_AUTH_DENIED:
    /* forward the packet to kernel */
    SYSLOG("Access denied");
    if (WriteFile (hOut, &packet,
                  sizeof (o_auth_packet),
                  &BytesWritten, NULL) == 0)
    {
        sprintf (message, "WriteFile failed : %d",
                  GetLastError ());
        SYSLOG (message);
        return (1);
    }
    if (BytesWritten != sizeof (o_auth_packet))
    {
        sprintf (message, "WriteFile wrote %d bytes instead of
                        %d", BytesWritten, sizeof(o_auth_packet));
        SYSLOG (message);
        return (1);
    }
}

```

```

        break;
    default:
        SYSLOG("Wrong reply packet");
        exit(2);
    }
    /* done */
    SYSLOG("Done");
    return(0);
}
/*
 * A sample server user authentication program.
 */
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <omapi.h>
#define PASSWORD    "simple-but-unsafe"
FILE    *outputFile = NULL;
#define SYSLOG(msg)    {                                \
    outputFile = fopen("authbe.out", "a");              \
    fprintf(outputFile, "(BE): %s\n", (msg));            \
    fclose(outputFile);                                \
}
/*
 * This is a bare-bone back-end authentication program
 * whose sole purpose is to demonstrate the usage of
 * Versant user authentication facility.
 * It simply reads in a plain-text password and verifies
 * it.
 */
int main (int argc, char **argv)
{
    o_auth_packet    packet;
    char    buff[30];
    DWORD BytesWritten, BytesRead;
    HANDLE hIn, hOut;
    char message[256];
    if (argc < 3)
        SYSLOG("Bad arguments");
    hIn = (HANDLE)atoi (argv[1]);

```

---

```
hOut = (HANDLE)atoi (argv[2]);
sprintf (message, "hIn = %d, hOut = %d", hIn, hOut);
SYSLOG(message);
SYSLOG("Start");
/* write an O_AUTH_READ packet to kernel and wait
 * for an reply */
packet.opcode = O_AUTH_READ;
packet.len = 0;
if (WriteFile (hOut, &packet, sizeof(o_auth_packet),
               &BytesWritten, NULL) == 0)
{
    sprintf (message, "WriteFile failed : %d", GetLastError ());
    SYSLOG (message);
    return (1);
}
if (BytesWritten != sizeof(o_auth_packet))
{
    sprintf (message, "WriteFile wrote %d bytes
    instead of %d", BytesWritten, sizeof(o_auth_packet));
    SYSLOG (message);
    return (1);
}
if (ReadFile (hIn, &packet, sizeof (o_auth_packet),
              &BytesRead, NULL) == 0)
{
    sprintf (message, "ReadFile failed : %d", GetLastError ());
    SYSLOG (message);
    return (1);
}
if (BytesRead != sizeof (o_auth_packet))
{
    sprintf (message, "ReadFile read %d bytes
    instead of %d", BytesRead, sizeof (o_auth_packet));
    SYSLOG (message);
    return (1);
}
if (ReadFile (hIn, buff, packet.len, &BytesRead,
              NULL) == 0)
{
    sprintf (message, "ReadFile failed : %d",
              GetLastError ());
```

```
        SYSLOG (message);
        return (1);
    }
    if (BytesRead != packet.len)
    {
        sprintf (message, "ReadFile read %d bytes
            instead of %d", BytesRead, packet.len);
        SYSLOG (message);
        return (1);
    }
    /* verify the reply packet */
    if (packet.len == strlen(PASSWORD) &&
        !strcmp(buff, PASSWORD, strlen(PASSWORD)))
    {
        SYSLOG("Access granted");
        packet.opcode = O_AUTH_GRANTED;
    }
    else
    {
        SYSLOG("Access denied");
        packet.opcode = O_AUTH_DENIED;
    }
    /* write the authentication result to kernel */
    packet.len = 0;
    if (WriteFile (hOut, &packet, sizeof(o_auth_packet),
        &BytesWritten, NULL) == 0)
    {
        sprintf (message, "WriteFile failed : %d",
            GetLastError ());
        SYSLOG (message);
        return (1);
    }
    if (BytesWritten != sizeof(o_auth_packet))
    {
        sprintf (message, "WriteFile wrote %d bytes
            instead of %d", BytesWritten, sizeof(o_auth_packet));
        SYSLOG (message);
        return (1);
    }
    /* done */
```

---

```
SYSLOG("Done");  
return 0;  
}
```



---

This Chapter explains the database creation basics and procedure.

The Chapter describes the following:

- Database Creation Basics
- Database Creation Procedure

## DATABASE CREATION BASICS

Personal and group databases can be created on a local disk or on a server machine. The process of creating a database is the same for servers and local disks.

## Prerequisites For Database Creation

### Database Root Directory

**Before you can create any databases, you need to designate a database root directory.**

For each new database, one subdirectory containing database specific information will branch from the database root directory.

The database root directory can be located anywhere. It does not have to branch from the Versant software root directory.

If you specify relative paths to the database volumes at the time each database associated with your installation is created (the default), you can easily move the location of the database root directory later.

The directory for each database must have the same name as the database.

Three volumes provide storage for each database: a system volume, a physical log volume, and a logical log volume. These volumes can be located anywhere. Additional storage volumes for the database can be created at any time.

To maximize performance, the system and data volumes, the physical log volume, and the logical log volume should be placed on separate disk drives. Placing the volumes on different drives minimizes disk seek time and thus maximizes throughput. If performance is not a critical issue, all volumes can be put on the same disk drive, which is simpler to do than separating them.

### Database Owner (DBA)

**Each database requires a directory for its configuration and system files. The owner of this directory is the "database administrator" or "DBA".**



---

Only the DBA of a database can perform certain administrative tasks on a database, such as adding or removing users.

After a database has been created, you can change its database administrator by changing the ownership of its database directory.

## Database System Identifier File (`osc-dbid`)

The name of the database system identifier file is `osc-dbid`.

**The database system identifier file, `osc-dbid` - must exist and be accessible to your machine before you can create a database.**

This file contains information about all databases that you and others might conceivably access during a particular database session.

The purpose of the database system file is to ensure that each database has a unique identifier number in a network system of databases. This is important, because the object model requires that each object have an identifier number unique among all other objects, regardless of database. When you create an object, the object is given an identifier number composed partly of the database identifier and partly of a number unique to the database.

You will not need to access the database system file when you create an object or connect to a database. You will need to access the database system file when you create a database or ask for information about any or all databases in a network system of databases.

In brief, the `osc-dbid` file keeps track of all database identifiers in a network of distributed databases: the database name, the database id, the creator, and the creation date. It is automatically updated for each new database.

If you want to add a new database to a distributed database system, the machine creating the database must be part of the existing network of Versant databases when the new database is created.

If the `osc-dbid` file for a distributed database system is destroyed, you cannot create new databases or list existing ones until you recover the `osc-dbid` file from a backup copy or create a new `osc-dbid` file. To create a new `osc-dbid` file, use the `dbid` utility.

**To see the information in `osc-dbid`, run the Versant `dblist` utility.**

For example, for the database `my_database`, the command

```
dblist -d my_database
```

...might return:

```
ID                = 4677
DB name           = my_database
creator           = silas
date created      = Thu Jul 14 22:51:50 2005
DB type           = personal
version           = 7.0.1.3
```

The location of the `osc-dbid` file is specified during the first installation of Versant in a particular database network.

**For more information on the `osc-dbid` file, See also “Database System Identifier File (`osc-dbid`)” on page 308 in "Chapter 5 - Directories and Files".**

## Database Volumes

The principal components of a database are:

### System Volume

The System Volume contains database system catalogs and provides data storage space.

### Storage Volumes

Optional Storage Volumes are added using the `addvol` utility to supply more storage space if the System Volume is not big enough.

### Physical Log Volume

The Physical Log Volume contains physical data information for logging and recovery.

### Logical Log Volume

The Logical Log Volume contains transaction undo and redo information for logging, recovery, and rollback.

For non-UNIX operating systems, database volumes must be files.

## Usage Notes for UNIX Users

- A database on UNIX can use either files or raw devices for the database volumes. If you are using files, they may be local or remote.
- If the database volumes are remote UNIX files, they may be accessed through NFS. However, we strongly discourage you from accessing database files through NFS, because the

---

NFS protocol does not guarantee that file writes are flushed to disk on invocation of a flush system call. This means that use of NFS could result in database corruption!

It is much better to create a database on the host where the file system is physically located and then access it with Versant `db@host` protocol rather than with NFS protocol. This allows a Versant server process to directly access the database files and will improve performance.

- If you want to use a UNIX raw device for a database volume, make sure that the partition you use does not include cylinder 0 or cylinder 1. When a UNIX partition is used for a raw device, all cylinders allocated to that partition will be used.
- If the UNIX partition is located on 'a' or 'c' ('c' implies the entire disk), the disk label can be overwritten.

The workaround is to make partition 'a' two cylinders long and then start partition 'b' at cylinder 2. You can then safely use partition 'b' for raw devices.

- If you use a UNIX raw device for the physical and/or logical log volumes, you may run out of space, because raw devices cannot be dynamically expanded by Versant. By contrast, if you use files for the log volumes, Versant will increase the log volume sizes when necessary. Of course, the log files cannot expand if their disk partition is full.
- In order to create database volumes on UNIX raw devices, you must make the person who will create the database the owner of the device, and give the database owner read-write permissions.

## Parameters for Creating Database Directories

While creating database directories, there are some mandatory parameters and some optional parameters which need to be set.

### Mandatory Parameters

**When creating database directories with `makedb`, you must specify the following parameters:**

All but the first parameter uses a default setting, if omitted.

**NOTE:-** These parameters cannot be changed after the database has been created.

#### Database name

The database name must be no longer than 31 characters. When referring to a remote database, you must append a site name using the `database@node` syntax. The site name `node` must be no longer than 223 characters.

#### Database type

The database type can be either `personal` or `group`. The default is `group`.

#### Database owner

The default owner is the `username` of the person creating the database.

### Optional Parameters

**When creating a database with `createdb`, you can set the following optional parameters in the database server process profile file `profile.be`.**

If you do not explicitly set any values, the defaults are used.

**NOTE:-** These parameters cannot be changed after the database has been created.

## Database size

The defaults are:

System Volume	128	megabytes
Physical Log Volume	2	megabytes
Logical Log Volume	2	megabytes

If the log volumes are files, they will expand as necessary during database use. The system volume does not grow. To increase storage for data after the database has been created, you can add more volumes by using the `addvol` utility.

The minimum size for each log volume is 256 kilobytes.

## Database volume file names

In the database server process profile file `profile.be`, you can specify paths and file names for the database volumes. The paths and names determine where the volumes are located and whether they are files or UNIX raw devices. The default is to use file volumes.

Each database must have a directory branching from the database root directory. In this directory environment files for the database are located. The name of a database directory must be the same as the database name.

The directory for a database can also hold the database volumes if they are files. If you use raw devices, of course, the volumes will not be located in the database.

If the path to the database volumes is specified in relative terms when the database is created, then the database volumes can be moved later. If the path is specified in absolute terms, the volumes can never be moved. The default is to specify a relative path to the database volumes.

**NOTE:-** You can change the following parameters after a database has been created, but as they are important, it is advisable that you decide on their values earlier.

Each of these parameters may be set in the database server process profile file `profile.be`:

## Locking

Whether locking is to be `ON` or `OFF`. The default for group databases is locking `ON`, and the default for personal databases is locking `OFF`.

## Logging

Whether logging is to be `ON` or `OFF`. The default for all databases is logging `ON`.

## DATABASE CREATION PROCEDURE

Following are the steps required to create a database.

1. **Set up database directories and default profiles**
2. **Optionally edit database profile**
3. **Create the database**
4. **Move a database**

**NOTE:-** Versant does not allow spaces in names, including login and database names.

These are explained in detail in the following sections:

### Step 1. Setting up the database directories and files

Before creating a database, you have to create a database directory, assign access privileges, and create database profiles.

You can use the Versant `makedb` utility to create the necessary default setup. The `makedb` utility will give the user an option to input a password to be associated with DBA.

You must create a database directory even if you are going to use UNIX raw devices for the database volumes. The directory contains essential database files, such as the database profile. The database directory must branch from the database root directory as specified during installation of the system software, and it must have the same name as the database that you want to create.

If you want to use UNIX raw devices for database volumes, run `makedb`, and after creating the raw device, manually set all access privileges. For example, in UNIX, you would use the `chown` and `chmod` commands.

If you are recreating a database previously removed with the `removedb` utility, you can use the old directories and profiles without having to rerun `makeprofile` or `makedb` unless you removed the database with `removedb -rmdir`.

You can set up a database either automatically by using the `makedb` utility or manually by performing the equivalent steps yourself.

---

## Automatic Setup

To perform an automatic setup of database directories and files, do the following.

**a. Login**

If you have write privileges for the home database directory `home_db_dir` which was created during the installation of the Versant software, then login with your user name. If you do not have write access to the databases directory `home_db_dir`, then login as root.

**b. Run `makedb`**

Create the database directories and files with `makedb`. The general syntax is:

```
makedb [options] dbname
```

If it does not already exist, the `makedb` utility will create a database directory named `dbname` per the options in the `options` parameter, and, if they do not already exist, create database support files. You must run `makedb` or create a database directory and profile files manually before creating a database with `createdb`.

For a remote database, append the node name to the database name using the syntax `dbname@node`.

For `options` you can substitute:

`-g`

Group database. This is the default option.

Group databases are accessible to many users at the same time.

`-p`

Personal database. Personal databases are accessible to one application at a time.

`-owner user`

Make `user` the owner of the database directory. The default is to use the current login name.

You can only make someone else the owner if you are logged in as the superuser.

`-cpprofile db`

Copy the profile file `profile.be` from the specified `db` directory to the directory for the new database.

A user can copy a database profile only if it is the owner of the database i.e., only the DBA is allowed to copy profiles.

This option works only if both databases are on the same machine.

If only a database name is used rather than a full path name, the `db` directory under the database root directory is searched.

`-nofeprofile`

Do not create an application process profile.

`-logging`

Turn transaction logging on.

`-locking`

Turn transaction locking on.

`-noprint`

Do not display any messages while the command runs.

The directory created will be:

`/dbroot/dbname`

A directory for the database that branches from the database root directory. The name of the directory will be the same as the database name specified as `dbname`.

Database support files that will be created, if they do not already exist, are:

`/.osc/dbname`

An application process profile file, located in the `.osc` directory branching from your home directory. This file contains operating parameters used when this database is a session database.

`profile.be`

A database server process profile file, located in the database directory. This file contains database creation and operating parameters.

`.lock`

A lock file, which indicates whether or not the database has already been started.

`.dbtype`



---

A database type file, which indicates whether this is a personal or group database.

For example:

```
makedb -p my_personal_database  
makedb our_group_database
```

c. Optionally create UNIX raw devices for the database.

If you want to use UNIX raw devices for the database, you must partition the disk drive and set ownership and access privileges for the raw devices. Also you must edit the server process profile `profile.be` to specify the raw devices when the database is created.

For example, assume that you have made the following raw device partitions:

```
/dev/rsh0g
```

For the System Volume

```
/dev/rsh0h
```

For the Physical Log Volume

```
/dev/rsh1b
```

For the Logical Log Volume

Then, to assign a user name to the raw devices, use the `chown` command:

```
chown username /dev/rsd0g /dev/rsd0h /dev/rsd1b
```

The specified `username` must be the user who will create the database and become its database administrator.

For an explanation on how to edit the server process profile `profile.be` to point to the raw devices, see the "Database Profiles" section.

## Manual Setup

If database directories have already been set up, you can run the `makeprofile` utility to create the application process and server process profile files. The general syntax is:

```
makeprofile [options] dbname
```

For a remote database, append the node name to the database name using the syntax `data-base@node`.

Options are:

`-cpprofile db`

Copy profiles from the specified `db` directory.

A user can copy a database profile only if it is the owner of the database i.e., only the DBA is allowed to copy profiles.

If only a database name is used rather than a full path name, the `db` directory under the database root directory will be used.

`-nofeprofile`

Do not create an application process profile.

`-logging`

Turn logging on.

`-locking`

Turn locking on.

`-noprint`

Suppress display messages while command runs.

For example, to make profiles for `myDB`:

```
makeprofile myDB
```

If you are recreating a database previously removed with the `removedb` utility, you can reuse the directories and profiles without having to run `makeprofile` or `makedb`.

## Step 2. Editing the database profile (profile.be)

You have to edit the database server process profile `profile.be` if you want to change the system volume size from its default value, use raw devices for the database volumes, or alter the default extent size.

The name of the database profile is `profile.be`, and it will be located in the database directory after you run the `makedb` utility.

Parameters for the sizes and names of database volumes cannot be changed after the database has been created.

- The default name of the System Volume is `system`, and its default size is 128 megabytes. Edit the `sysvol` entry in `profile.be` to change the size of the System Volume.
- The default name of the Physical Log Volume is `physical.log`. The default initial size is 2 megabytes. Edit the `plogvol` entry to change the size of the Physical Log Volume.
- The default name of the Logical Log Volume is `logical.log`. The default initial size is 2 megabytes. Edit the `llogvol` entry to change the size of the Logical Log Volume.
- The default type for database volumes is files.
- If files are used for the log volumes, the log volumes expand as needed; the system volume does not change. If you specify a size for the log volumes and the volumes are files, then the parameter will be used to pre-allocate space, but the files will still expand as needed.
- If you use raw devices for database volumes, you must specify a path name and size for each, because there is no default location for a raw device. If the log volumes are on raw devices they will not extend.
- The default number of pages per extent on the system volume is 2. To change the number of pages per extent, edit the `extent_size` entry in the `profile.be` file.

There are other entries in the server process profile. They are performance related and can be changed before or after creating a database.

**For more information, See also “Server Process Parameters” on page 84.**

For a standard installation, the default system volume, physical log volume, and logical log volume entries are:

<code>sysvol</code>	<code>128M</code>	<code>system</code>
<code>plogvol</code>	<code>2M</code>	<code>physical.log</code>
<code>llogvol</code>	<code>2M</code>	<code>logical.log</code>

The general syntax for each entry is:

<code>sysvol</code>	<code>size</code>	<code>[/path/]system</code>
<code>plogvol</code>	<code>size</code>	<code>[/path/]physical.log</code>
<code>llogvol</code>	<code>size</code>	<code>[/path/]logical.log</code>

where:

`size`

The volume size in bytes. Kilobytes can be specified with `k` or `K`, and megabytes can be specified with `m` or `M`. You can specify any size you want for the volume size.

`/path`

The path to the volume. It can be specified in relative or absolute terms. If the first character is `/`, then the path is absolute. The path for each volume can be different.

`system`

Name of the system volume.

`physical.log`

Name of the physical log volume.

`logical.log`

Name of the logical log volume.

For raw devices, if you specify a size smaller than the actual raw device, only `size` space will be used. If you specify a size larger than the actual raw device, the `createdb` utility will catch the error and raise an exception.

For example, to specify raw devices, a system volume of 40 megabytes, a physical log of 2 megabytes, and a logical log of 5 megabytes:

```
sysvol      40M /dev/rsd0g
plogvol     2M  /dev/rsd0h
llogvol     5M  /dev/rsd1b
```

If only volume names are specified in the server process file, Versant will use the specifications in the `/etc/.oscxyyz` file or in environment variables to expand the volume names into full path names. The actual name of the `.oscxyyz` file depends on the release, where `xx` is the major release number, `yy` is the maintenance release number, and `zz` is the minor release number. For example, for Release 7.0.1.3 the file would be `.osc070001`.

For example, if your database name is `dbname` and your `/etc/.osc070001` file specifies your software root as:

`/usr/local/versant`

and your database root is set to the installation default location:

---

```
/usr/local/versant/db
```

and you specify your volumes as:

```
sysvol      40M system
llogvol     2M logical.log
plogvol     5M physical.log
```

then the system will expand your entries to

```
sysvol      40M /usr/local/versant/db/dbname/system
llogvol     2M /usr/local/versant/db/dbname/logical.log
plogvol     5M /usr/local/versant/db/dbname/physical.log
```

### Step 3. Creating the database

After setting up the database directories, optionally creating raw devices, setting database access privileges, and optionally editing the server process profile, you can create a database by running the Versant `createdb` utility.

The `createdb` utility will create the database including storage and log volumes and update the network database identifier file `osc-dbid` to register the new database as part of the database system.

**The general syntax for `createdb` is:**

```
createdb [options] dbname
```

The `createdb` utility will create, format, and initialize a new database with the name `dbname` and either make the new database a part of an existing distributed database system or start a new database system. The database name `dbname` must be unique for the distributed database system it will belong to.

You must be the owner of the database (the database administrator) or the super user to run this utility.

For a remote database, append the node name to the database name using the syntax `data-base@node`.

Options that may be specified in the `options` parameter are:

```
-i
```

Reserve space for the system volume.

Reserving space can prevent you from running out of space at runtime. For example, suppose you set the size of the system volume as 100 megabytes in your profile, but you really have only 60 megabytes physically available. In this case, if you use `-i`, you will immediately get an out-of-space error, but if you do not use `-i`, you will get an error at runtime as the system volume dynamically expands.

If you have defined multiple storage volumes and the physical space in one volume is less than the logical space, you will get an out-space-error even though there is space available in the next volume.

Pre-allocating space may improve performance, because space will not have to be dynamically allocated. However, pre-allocating space will slow down the process of creating the database, and because the space is used immediately, it will prevent it from being used for other purposes.

The `-i` option has no effect on UNIX raw devices.

Space for log volumes is always pre-allocated according to the parameters set for the `plogvol` and `llogvol` size specifications.

`-il`

Reserve and pre-format file system space for the logical log volume and the physical log volume.

`-noprint`

Suppress display messages while running.

The `createdb` utility will update the `osc-dbid` file to register the new database with the existing distributed database system. The file `osc-dbid` containing the path and name of all databases in the system must be visible from your machine when you run `createdb`.

### Examples of using `createdb`:

```
createdb mydb
createdb -i mydb
```

Components of the newly created database will be:

### System volume

A system volume is needed for catalog information and data storage, with a name, location, size, and device according to the specification in the server process profile file `profile.be`.

---

The default is a UNIX file of 128 megabytes named `system` whose location for a database named `dbname`, is:

```
/usr/local/versant/db/dbname/system
```

## Physical log volume

A physical log volume is needed for physical data information related to logging and recovery, with a name, location, size, and device as specified in the server process profile file `profile.be`.

The default is a UNIX file of 2 megabytes named `physical.log` whose location for a database named `dbname`, is:

```
/usr/local/versant/db/dbname/physical.log
```

## Logical log volume

A logical log volume is needed for transaction undo-redo information related to logging and recovery, with a name, location, size, and device as defined in the server process profile file `profile.be`.

The default is a UNIX file of 2 megabytes named `logical.log` whose location for a database named `dbname`, is:

```
/usr/local/versant/db/dbname/logical.log
```

## .sharemem

A shared memory file, which contains a shared memory identifier.

## Step 4. Moving a database

After you have created a database, you can move the volumes of a database to a different directory, with some restrictions.

For example, on a particular machine, you might decide that you want to locate the system software on one disk and the databases on another disk. Or you might want to use the system software on a server while maintaining databases on a local disk.

**CAUTION:** You should not move a database after it has been created unless you are an experienced user.

## Moving just the data volumes

You can always move the data volumes by editing the system profile file (`profile.be`), even if you specified an absolute path when you created them. However, this is not true of the system, physical log, and logical log volumes.

To move the data volumes:

1. Move the data volumes to a new location.
2. In the server profile file, `profile.be`, create or change the `datavol` entry for each data volume you have moved.

**For more information refer to “datavol” on page 99 in "Chapter 3 - Database Profiles".**

## Moving all volumes

You can create a database and later move all database volumes to a different directory if, before database creation, the volumes were specified in relative terms rather than absolute terms. This can occur in any of three ways:

- You originally placed the volumes under the database directory.
- Even if you specified the location of the database volumes in absolute terms, if they were placed under the database directory, they are internally stored in relative terms... which means that they can be moved at a later time.
- You created the database using the defaults.
- By default, volumes are stored under the database directory.
- Before database creation, you edited the server process profile file in such a way that the volume paths were specified in relative terms.

To move all volumes to a new location, do the following.

1. Move the database volumes to a new location, preserving their relative relationships.
2. Reset the location parameter `VERSANT_DB` to indicate the new database root.

**For more information on location parameters, refer to Chapter 5 “Directories and Files” on page 281.**



---

This Chapter gives us an brief insight about the Database Profile Parameters.

The Chapter covers the following in detail:

- Application Process Parameters
- Server Process Parameters

## APPLICATION PROCESS PARAMETERS

### Overview

When you use the `makedb` or `makeprofile` utilities, if an application profile file does not already exist, Versant will create an application process profile with the same name as the database you are creating.

When you start a database session and specify a database to be used as the session database, Versant reads the application profile for that database and uses the parameters it finds to set the operating environment for the application.

At any time, you can edit the contents of an application profile to change the operating environment for your application process. The parameters are read each time a session starts. The existence of an application process profile file is optional: if one does not exist, default values will be used.

It should be noted that different users use different application profiles while connecting to the database even if they are running the same program because the application process profile will be read from different locations.

If you want the changes in application process parameters to be reflected for all the users, you need to distribute the modified application process profile file to all the users of the database.

### On Windows

When you install, Versant will create a subdirectory named `osc` under your home directory. This directory will contain application process profiles for the databases that you create.

For example, for a database named `dbname`, the application process profile is:

```
%HOMEDRIVE%%HOMEPATH%\osc\dbname
```

### On UNIX

When you install, Versant will create a subdirectory named `.osc` under your home directory. This directory will contain application process profile files for the databases that you create.

For example, for a database named `dbname`, the application process profile is:

```
$HOME/.osc/dbname
```

When setting parameters involving bytes, you can specify kilobytes with `k` and megabytes with `M`.

To create a comment line in the profile file, you must use the pound sign `#` as the first character of the line.

The application process profile can understand and use the following parameters.

### Functional parameters

Parameter	Description
<code>alias</code>	Aliases for local and remote databases.
<code>connect</code>	Automatic connections to group databases each time a session starts.
<code>genericObject</code>	Turn on or off the use of the generic version object.
<code>loose_schema_mapping</code>	Allow inconsistent class definitions.
<code>max_processes</code>	The maximum number of processes within the system.
<code>parallel_write</code>	Write objects in parallel or serial to a pair of FTS databases.
<code>signal_block</code>	Block unwanted signals, such as Ctrl-C, that might otherwise damage shared memory. If you run your application with a single process, you may need to turn this parameter on.

### Tuning parameters

Parameter	Description
<code>dbconnect_timeout</code>	Overrides the Operating System TCP/IP driver <code>connect()</code> timeout settings.
<code>estimated_connections</code>	Set the maximum number of database connections that can be made in a database session.
<code>estimated_objects</code>	Set the estimated number objects that will be accessed in a session.
<code>heap_size</code>	A pre-allocation hint for the size of the application heap, which includes cached objects, the cached object descriptor table, and other data structures.
<code>heap_size_increment</code>	Expansion increment of application heap.
<code>max_objects</code>	Set initial cod table size.

`swap_threshold`

Swap threshold. If less than this amount of the application heap is used, object swapping will not be attempted by the system.

The Functional and Tuning parameters are explained in detail as follows:

## Functional Parameters

Following are the Functional parameters.

In the parameter descriptions below, `database_name` refers to the application process file belonging to a database of the same name.

### alias

```
alias alias_name database_name[@node]
```

Set the alias `alias_name` for the database `database_name`.

For a remote database, append the node name to the database name using the syntax `database@node`.

Aliases are kept throughout a session. Each alias must be entered on a separate line. Setting aliases is optional.

Aliases have two purposes: convenience and performance tuning.

**Convenience:** The primary purpose of aliases is to simplify application syntax and to allow later changes of alias references without having to change applications that use the aliases.

When an application uses a remote database, specify the remote database name in such a way that it indicates that the database is on another machine. You do this with `database_name@location` syntax. This syntax can either be used in each reference made to the remote database in an application or by creating an alias in the Application Process parameter file for the database you are using as your session workspace.

For example, if your application uses two group databases named `groupdb_1` and `groupdb_2` located on machines `machine_1` and `machine_2`, you can add the following lines to the file `$HOME/.osc/personaldb`:

```
alias gdb1 groupdb_1@machine_1
```

---

```
alias gdb2 groupdb_2@machine_2
```

After setting these aliases, you can then refer to the group databases in your application with the names `gdb1` and `gdb2`. If you later change the location of either database or decide to use different group databases in your application, you only have to change the aliases rather than editing your program.

**Performance tuning:** Aliases allow the use of different database parameters for one database depending upon the application using it. For example, for one application you might want a large heap size, while for another one you might want a small heap size.

To use of several sets of database parameters for one database, first create several sets of profiles with differing names. For example, you might create profiles named `"profile_one"` and `"profile_two"`.

To specify which application process file to use for a particular application, use database aliases. In each of the profiles you create, specify an alias for the database that you want to use as your session workspace.

Suppose, you have a database whose real name is `db` and you have two application process parameter files: `$HOME/.osc/db1` and `$HOME/.osc/db2`. The `db1` file should have the alias entry `"alias db1 db"`, and the `db2` file should have the entry `"alias db2 db"`. Then, in your program, if you specify `"beginsession(db1,...)"`, the database `db` will be started using the entries in the application profile file `db1`. If you specify `"beginsession(db2,...)"`, the values in the file `db2` will be used.

If there is no application process parameter file with the name specified in a "begin session" method, then a session is started in the database of the same name using default values of the parameters.

## connect

```
connect group_db access_mode
```

Automatically connect to the group database `group_db` when you begin a session.

Access modes are specified as:

### **READ\_ONLY**

Read only.

### **READ\_WRITE**

Read and write.

### WRITE\_ONLY

Write only.

You can make multiple `connect` entries to connect automatically with multiple group databases. Specify remote databases with `db@node` syntax.

Connections to databases made using the `connect` parameter can be disconnected by calling a "disconnect database" routine from within a session. Otherwise, the connections will be terminated when you end the session.

For example, if you wish to connect to a group database named `groupdb1` on `machine1` and `groupdb2` on `machine2` every time you begin a session, add the following statements to your application process parameter file:

```
connect group1@machine1 READ_WRITE
connect group2@machine2 READ_WRITE
```

## genericObject

`genericObject` ON | OFF

Turn use of the generic object off and on.

The default is OFF. This setting is not used any longer, is left for backward compatibility.

## loose\_schema\_mapping

`loose_schema_mapping` *state*

Turn loose schema mapping ON or OFF.

The default state is OFF.

When the loose schema mapping is `off`, an application compiled with an old class definition can continue to use a database even if class definitions differ.

When the loose schema mapping is `on`, whenever an application accesses a class in a database, Versant compares the database definition of the class with the definition expected by the application. If the database and application definitions of the class differ, an exception is raised.

If `loose_schema_mapping` is `on`, the following behavior occurs.

---

If attribute is same in application and database...

Create object — OK.

Read object — OK.

Write object — OK.

If attribute exists in database only...

Create object — Attribute is set to zero when object is written to database.

Read object — Attribute is not seen.

Write object — Attribute is not modified when object is written to database.

If attribute exists in application only...

Create object — Attribute is not written to database.

Read object — Attribute is set to zero.

Write object — Attribute is not written to database.

If attribute type has been changed...

Create object — Exception is raised: "types do not match."

Read object — Exception is raised: "types do not match."

Write object — Exception is raised: "types do not match."

If `loose_schema_mapping` is `off`, the following behavior occurs.

If attribute is same in application and database...

Create object — OK.

Read object — OK.

Write object — OK.

If attribute exists in database only...

Create object — Exception is raised: "signature mismatch" or "wrong layout."

Read object — Exception is raised: "signature mismatch" or "wrong layout."

Write object — Exception is raised: "signature mismatch" or "wrong layout."

If attribute exists in application only...

Create object — Exception is raised: "signature mismatch" or "wrong layout."

Read object — Exception is raised: "signature mismatch" or "wrong layout."

Write object — Exception is raised: "signature mismatch" or "wrong layout."

If attribute type has been changed...

Create object — Exception is raised: "types do not match."

Read object — Exception is raised: "types do not match."

Write object — Exception is raised: "types do not match."

## parallel\_write

`parallel_write ON | OFF`

Use a write optimization that writes objects in parallel, to a group of databases and thereby improves the overall write performance.

The default option for this parameter is `ON`.

It is important to understand how object writes normally work in order to understand the influence of this parameter.

The write consists of two main phases, first, writing the objects to the destination database server and second, waiting for a confirmation from the server that the write was successful. The first phase of writing to the server is resource intensive as it involves logging and other activities and the client usually blocks while waiting for a response from the server. It is more efficient if this "blocking" time is better utilized to write out the other objects to other databases. This is the fundamental design philosophy behind parallel write.

If `parallel_write` is `OFF` and an application explicitly or implicitly flushes / writes objects to multiply connected databases in a session then both phases of the write is performed on one database at a time.

If `parallel_write` is `ON` then the VOD client "Object Manager" performs the first phase of the write on all the respective databases and then the second phase is performed.

Enabling the `parallel_write` feature also improves VOD Veddung performance, since Veddung also involves multiple connections to databases (One explicit connection to a primary database and an implicit connection to the secondary database).



## signal\_block

signal\_block ON | OFF

Blocks unwanted signals. The default is OFF.

When signal\_block is ON, the following signals are blocked:

```
SIGHUP SIGINT SIGQUIT SIGPIPE SIGALRM SIGTERM SIGURG
SIGSTOP SIGTSTP SIGCONT SIGCHLD SIGIO SIGWINCH SIGLOST
SIGUSR1 SIGUSR2
```

You may need to turn this parameter ON if you run your application with a single process. In single process mode, an application makes calls directly to its session database, so if a user hits Control-C, for example, during a critical moment, the database could become inconsistent.

## Tuning Parameters

Versant has many parameters that you can configure to optimize database performance.

Following are the Tuning parameters:

### dbconnect\_timeout

dbconnect\_timeout number

This parameter can be used to override the Operating System TCP/IP driver timeout settings of connect() calls. The values accepted for this parameter is an integer in seconds. Non-integer values will result in error OM\_PROF\_SYNTAX.

A client issues a connect() whenever a o\_beginsession() or o\_connectdb() is called. The session database settings for dbconnect\_timeout will be used to connect to all the databases in the session.

The default value is -1. When the default value is set, the TCP/IP driver settings for connect() timeout is used. Any non-negative value will override the TCP/IP driver timeout settings of connect() calls. The value is limited to the maximum value of a signed integer. The value 0 indicates zero timeout and is not recommended as the connection would most certainly timeout. All negative values indicate a value of -1.

## estimated\_connections

`estimated_connections` number

Set the maximum number of database connections that can be made in a database session.

In the current release, despite the usage of "estimated" in its name, the `estimated_connections` parameter sets a hard limit on the number of concurrent connections that can be made.

The default value of this parameter is 4.

This parameter is useful in reducing memory requirements.

## estimated\_objects

`estimated_objects` number

Explicitly specify the estimated number of objects that you will access in a database session.

The default value is 32K objects.

You can set this number higher if you will access more than 32K objects in a session.

This parameter affects the size of system information tables, such as the cached object descriptor table, that are created when you start a database session. If your application creates and uses multiple, concurrent database sessions and, in each session, accesses only a few objects, you can set this parameter low to reduce memory usage. On the other hand, if you are using a single session at a time and accessing a large number of objects in the session, you can set this parameter high to improve performance.

**See also** `heap_size`.

## heap\_size (application)

`heap_size` size

A pre-allocation hint for the size in bytes of the application heap. The applications heap contains cached objects, the cached object descriptor table, and other data structures.

If your swap space is too small, you may run out of swap space while your application is running. If you know that a session will require a lot of memory, you can pre-allocate space for the

---

application process heap by setting the initial heap size. If more memory than the initial amount specified by this parameter is needed, the heap size is dynamically increased as needed.

If your application requires more heap space than is available, an error will indicate that you have run out of swap space.

Normally you do not have to set this parameter since the application process heap is expanded dynamically as needed, and Versant can determine an appropriate initial value.

**See also “heap\_size (database)” on page 110, in server process parameter.**

## heap\_size\_increment (application)

`heap_size_increment size`

Set the size by which the application process heap will be expanded as needed.

The default increment is 1 megabyte.

**See also “heap\_size (application)” on page 80, in application process parameter.**

## max\_objects

`max_objects size`

Specify a minimum initial size in bytes for the cached object descriptor table. This initial size also becomes the increment by which the size of the cached object descriptor table increases as needed.

The default value is 32K.

The cached object descriptor table is dynamically allocated and will increase in size as needed.

This parameter is relevant if you are accessing only a few objects in a session, which might be the case when, say, you are using a large number of concurrent sessions. If you are going to access only a few objects, then you might want to decrease this value in order to reduce memory costs. If you are going to access a large number of objects, you might want to increase this value in order to improve performance.

**See also** `heap_size`.

## swap\_threshold

`swap_threshold size`

Specify the point at which object swapping begins.

Object swapping is the process in which objects residing in an application object cache are returned to their databases. When the object cache tries to grow beyond the swap threshold size, object swapping will occur.

The default swap size is 512 megabytes.

The swap threshold is a hard limit on the size of the cache, so swapping will begin as soon as the size of the cache exceeds the value of the swap threshold parameter.

An object chosen for swapping will be written to its database if it has been marked as dirty or discarded if it is clean. When objects which have been swapped out are accessed, they must be retrieved from the server.

Object swapping is not as good as keeping the objects cached in physical memory, but it is far better than virtual memory thrashing.

The optimal value of this parameter depends on your application and environment. If objects are always referenced in the same order, then objects which are used together are stored in contiguous memory addresses, and virtual memory will perform well. In this case, `swap_threshold` should be set much higher than physical memory, so that virtual memory is used instead of object swapping.

If, on the other hand, objects are accessed in an unpredictable order, object swapping is preferred and the parameter should be set low enough to prevent thrashing.

In any event, you should set the swap threshold high enough so that as much of physical memory as possible is used.

The following error messages may indicate that you are running out of swap space:

Error Code	Error Message	Description
0054	VSL_MEM_NOMEM	Out of process memory.
1100	SM_E_NOMEM	Out of back-end heap memory.
3002	NET_ENOMEM	Unable to allocate memory.
4161	OM_HEAP_NOMEM	Out of front-end heap memory.
6009	SCH_NOMEM	No heap memory in schema management.

---

**NOTE:-** This parameter affects the logical log.

A low swap threshold can cause the logical log file to grow, because the writing of dirty objects to a database generates logical log records. Consider the following scenarios.

### **Scenario 1:**

An object is modified only once during the course of a transaction and the `swap_threshold` value causes it to be swapped out at some point. In this case, there is no relative impact on the logical log, because a record would be written to the logical log anyway when the transaction committed.

### **Scenario 2:**

An object is modified several times during the course of a transaction, but is swapped only once at the end of all these modifications. In this case, there is still only one group-write operation, and one set of logical log records for the object.

### **Scenario 3:**

This is the worst case. The object is modified several times during a transaction and also swapped several times because of the swap threshold kicking in. In this case, there is "thrashing" of the object cache. Now there would be several logical log records for the same object, with a new set of logical log records being generated each time a swap occurred. If the object is a "hot" object, which needs to be referenced and updated frequently, it could cause the logical log file to grow dramatically and exceed its limits.

Scenario 3 shows that if your cache is too small as defined by the `swap_threshold` parameter, you may see unexpected growth in your logical log file. The fix is to either increase the size of your object cache by increasing the swap threshold value or else to increase the space available for your logical log file to grow.

As a rule of thumb, we recommended that if you have a large database application, you should allocate at least 96 megabytes of swap space.

## SERVER PROCESS PARAMETERS

### Overview

When you use the `makedb` or `makeprofile` utilities, if a server process profile file does not already exist, Versant will create one for the specified database.

When a database starts up, the database server process reads the server process profile for that database to determine the location of the database volumes and to set the database operating environment. If a server profile does not exist for a database, you cannot start that database.

### Server process profile location

When you create a database, Versant will create a subdirectory for the database under your database home directory. The server process profile will be created in the database subdirectory and be named `profile.be`.

#### On Windows

For example, for a database named `mydb` and a database root directory of `c:\versant\db`, the location of the server profile will be:

```
c:\versant\db\mydb\profile.be
```

#### On UNIX

For example, for a database named `mydb` and a database root directory of `/usr/local/versant/db`, the location of the server profile will be:

```
/usr/local/versant/db/mydb/profile.be
```

When setting parameters involving bytes, you can indicate kilobytes with `k` or `K` and megabytes with `m` or `M`.

To create a comment line in the profile file, you must use the pound sign `#` as the first character of the line.

## Server process profile parameters types

The different types of Server process profile parameters are:

**Database Creation:** Versant uses these parameters only when you create a database with `createdb`. If you change these parameters after you create the database, the system just ignores your changes.

Parameter	Description
<code>sysvol</code>	The maximum size of the system volume.
<code>plogvol</code>	The size of the physical log volume.
<code>llogvol</code>	The size of the logical log volume.
<code>extent_size</code>	The number of pages per extent on the system volume.

**Functional parameters:** Versant uses these each time you start the database. You can edit these parameters to affect the performance of your database.

Parameter	Description
<code>async_buffer_cleaner</code>	How cache flushing is performed.
<code>async_logger</code>	How log flushing is performed.
<code>commit_delete</code>	Whether object deleted at commit
<code>commit_flush</code>	Whether buffer is flushed after a commit.
<code>custom_be_plugins</code>	Location of VAT plugins
<code>datavol</code>	Location of storage volumes.
<code>event_daemon</code>	The event message daemon
<code>event_message_mode</code>	The event message mode
<code>event_msg_transient_queue_size</code>	The transient event message queue size.
<code>event_registration_mode</code>	Event registration mode.
<code>locking</code>	Turn short locking on or off.
<code>logging</code>	Turn logging on or off.
<code>startup_script</code>	Specify executable name that will be run at database startup.

<code>versant_be_dbalogginglevel</code>	Set DBA logging level as 0 or 1 for a database.
<code>versant_be_dbalogginglevel</code>	File used to log DBA utility activity.
<code>versant_be_logfile</code>	Specify location of the per-database LOGFILE.
<code>be_syslog_level</code>	Control system logging

**Tuning parameters:** Versant uses these each time you start the database. You can edit these parameters to change the operating parameters of your database.

Most of these parameters are optional, as the default values are appropriate for most situations. However, the parameter `class` is crucial to databases containing a large number of classes - if more than the default number of 480 classes will be accessed during the time a database is running, you will need to specify a new value for `class`.

Parameter	Description
<code>bf_dirty_high_water_mark</code>	When the cleaner thread starts flushing dirty pages.
<code>bf_dirty_low_water_mark</code>	When the cleanup thread stops flushing dirty pages.
<code>class</code>	The number of cached user defined classes.
<code>db_timeout</code>	Database time-out.
<code>heap_size</code>	A hint for the initial size of server heap.
<code>heap_size_increment</code>	Size by which server heap can be expanded.
<code>index</code>	The number of cached user defined indexes.
<code>llog_buf_size</code>	Size of the logical logging buffer in bytes.
<code>lock_batch_size</code>	Number of lock requests to be batched together
<code>lock_wait_timeout</code>	Time to wait for a lock.
<code>max_page_buffs</code>	The number cache buffers.
<code>multi_latch</code>	Database latching behavior.



obe_port_begin	Specify starting port number for connections. This parameter is no longer supported by VERSANT, please use the parameter VERSANT_SERVER_PORTS instead.  <b>For more details refer to “VERSANT_SERVER_PORTS” on page 346 in "Chapter 6 - Configuration Parameters".</b>
obe_port_end	Specify ending port number for connections. This parameter is no longer supported by VERSANT, please use the parameter VERSANT_SERVER_PORTS instead.  <b>For more details refer to “VERSANT_SERVER_PORTS” on page 346 in "Chapter 6 - Configuration Parameters".</b>
plog_buf_size	Size of the physical logging buffer in bytes.
polling_optimize	Strategy for FTS recovery.
spin_count	Spin lock busy wait loop count
transaction	The maximum number of concurrent transactions.
user	Initial size of user names list.
volume	The number of cached data volumes.
logvolmaxsize	The maximum size to which the logical and physical log volumes can grow

**Statistical parameters:** Versant can collect statistics and vary assertion levels using these parameters.

Parameter	Description
stat	Turn collection of connection and database statistics on or off.
assertion_level	Type of assertions.

**Tracing parameters:** Versant has tracing facilities. You can set the number of traces and the file in which the system stores the traces by using these parameters.

Parameter	Description
<code>trace_comps</code>	Database components to be traced.
<code>trace_entries</code>	Number of trace entries to be maintained in the trace file.
<code>trace_file</code>	Name of the trace file.

**Authentication parameter:** Versant has authentication parameters used for user authentication.

Parameter	Description
<code>Authentication_program</code>	Set location of user authentication program.
<code>be_permit_pre70_clients</code>	Permit Pre7.0 clients to connect

**Event notification parameters:** Versant can save event registrations and generated events by using these parameters.

Parameter	Description
<code>event_registration_mode</code>	Persistent or transient registrations.
<code>event_msg_mode</code>	Persistent or transient messages.
<code>event_daemon_notification</code>	Notification type when event daemon goes down.

**Database space parameters:** Versant has parameters to know the space of the database.

Parameter	Description
<code>Out_of_space_warning_thres hold</code>	Threshold value that triggers generation of system messages
<code>out_of_space_warning_thres hold_increment</code>	Indicates the increment at which a warning message should be generated

## Database Creation Parameters

Versant uses these parameters only when you create a database with `createdb`.

If you change these parameters after you create the database, the system just ignores your changes.

Parameter	Description
<code>extent_size</code>	The number of pages per extent on the system volume.
<code>sysvol</code>	The maximum size of the system volume.
<code>plogvol</code>	The size of the physical log volume.
<code>llogvol</code>	The size of the logical log volume.

### extent\_size

`extent_size number_of_pages`

The `extent_size` parameter specifies the number of pages per extent on the system volume.

The default is 2 pages: `extent_size 2`

The default is appropriate for most applications and should be increased only if your objects are very large.

An extent is the minimum unit of allocation to segments. The number of pages per extent tunes time versus space tradeoffs.

In general, setting a low number of pages per extent causes less space to be wasted by fragmentation and decreases the initial size of an empty database. It also makes it more likely that, over time, objects will be broken up and placed in different physical locations on the disk, thus slowing down access to the data.

In general, setting a high number of pages per extent decreases the likelihood that objects will be broken up and placed in different physical locations on disk, but it also increases usage of disk space.

**See also “sysvol” on page 92.**

## llogvol

`llogvol size name`

The size and location of the logical log volume used for transaction undo-redo information for logging and recovery.

### Parameters are:

#### **size**

An optional allocation size.

The default size is 2 megabytes. The minimum size is 256K. If you specify a value smaller than the minimum, 256K will be allocated.

If the logical log volume is a file, it will be expanded if necessary. If the logical log volume is expanded beyond the stated soft limit out of necessity, it will be shrunk back after recovery has been done.

#### **name**

Either the absolute or relative path (including name) of the logical log volume.

The default is to use the name `logical.log` and then search for path specifications.

If you specify an absolute path, you will not be able to move the database once it has been created.

The volume path can be either a file path or a raw device path. For raw devices, you must specify an absolute path.

You cannot change the name for the logical log volume once a database has been created.

When a transaction commits, a commit log record is written to the logical log file and the state of the transaction is guaranteed. In case of a system crash, the committed transaction is reconstructed by re-executing all of its actions from the log.

If logging is `ON`, the logical log file should be at least the sum of the sizes of all objects involved in a transaction. The larger the log file, the less frequently the system needs to pack the file when it is full. The logical log file should usually be larger than the physical log file.

For raw devices, if logging is `ON`, a guideline for setting the minimum size of `llogvol` is:

$$\text{guideline} = 32K + (2 * (\text{max\_o} + 100)) * \text{num\_o\_per\_tr} * \text{num\_tr}$$

---

In the above formula, substitute for `max_o` the size in bytes of the largest object in the database; substitute for `num_o_per_tr` the largest number of objects that will be involved in a transaction; substitute for `num_tr` the maximum number of concurrent transactions.

## plogvol

`plogvol size name`

The size and location of the physical log volume used for physical data information related to logging and recovery.

### Parameters are:

#### **size**

An optional allocation size. The default is 2 megabytes. The minimum size is 256K. If you specify a value smaller than the minimum, 256K will be allocated.

If the physical log volume is a file, it will be expanded if necessary. If the physical log volume is expanded beyond the stated soft limit, it will be shrunk back when a checkpoint occurs.

#### **name**

Either the absolute or relative path (including name) of the physical log volume.

The default name is to use `physical.log` and then search for path specifications.

See the "Run Time Environment" section following in this chapter.

If you specify an absolute path, you will not be able to move the database once it has been created.

The volume path can be either a file path or a raw device path. For raw devices, you must specify an absolute path.

You cannot change the name for the physical log volume once a database has been created.

If logging is ON, the physical log file should be at least twice the size of the largest object. A guideline for setting the minimum size of `plogvol` is:

$$\text{guideline} = 2 * (\text{max\_o\_size} + 1\text{K}) + 32\text{K}$$

In the above formula, substitute the size in bytes of the largest object in the database for `max_o_size`.

### sysvol

*sysvol size name*

The size and location of the system volume that contains database system catalogs and objects.

#### Parameters are:

##### **size**

The size of the system volume.

With `extent_size` = 2, the default size is 128 megabytes.

The absolute maximum is platform dependant.

### Large File Support

#### On Unix

This is applicable to Solaris 2.6 and above.

The system volume and the additional data volumes, can have a maximum size of 65535M

( just under 64GB).

#### On Windows

The system volume and the additional data volumes can have a maximum file size of 4095M (just under 4GB).

The `extent_size` is internally increased to the appropriate value to fit these requirements. For example if the `extent_size` specified in the profile is 2 and the system volume is size of 65535GB. The extent size is internally increased to 64.

##### **name**

Either the absolute or relative path (including name) of the system volume.

The default name is to use `sysvol` and then search for path specifications.

If you specify an absolute path, you will not be able to move the database once it has been created. Volume paths can be either a file path or a raw device path.

You cannot change the size or name for the system volume once a database has been created. To add space to a database, you must either add additional database volumes (with the `addvol` utility) or remove and recreate the database with a larger value specified for the `sysvol` parameter.

The system volume will use only as much space as needed, unless it is a raw device or the `-i` parameter is used with `createdb`.

A guideline for setting the size of `sysvol` is:

```
guideline = min_size + (1.25*(avg_o_size+40) * num_objects)
```

where `avg_o_size` is the average size of objects in the database and `num_objects` is the number of objects.

The minimum and maximum database volume sizes vary depending upon the value set for the extent size:

<b>extent_size</b>	<b>Minimum db volume size</b>	<b>Maximum db volume size</b>
1	1800K	1023M
2 (default)	1800K	2047M
3	1800K	3071M
4	1851K	4095M

The `createdb` and `addvol` utilities will raise an error if you create or add a database volume smaller than the minimum sizes.

If you run out of space in the system volume, you will see the following error message during the execution of an application program:

```
1083 SM_E_OUT_OF_VOL_SPACE all volumes exhausted
```

When you get this message, you should use the `addvol` utility to create additional database space.

Occasionally, you may get a panic message if a database is out of space inside a critical region. The panic message may be error message 1083 as shown above.

You may specify a raw device for the system volume. The path must be an absolute path.

## Functional Parameters

**Functional parameters:** Versant uses these each time you start the database.

You can edit these parameters to affect the performance of your database.

Parameter	Description
<code>async_buffer_cleaner</code>	How cache flushing is performed.
<code>async_logger</code>	How log flushing is performed.
<code>commit_delete</code>	Whether object deleted at commit
<code>commit_flush</code>	Whether buffer is flushed after a commit.
<code>custom_be_plugins</code>	Location of VAT plugins
<code>datavol</code>	Location of storage volumes.
<code>event_daemon</code>	The event message daemon
<code>event_message_mode</code>	The event message mode
<code>event_msg_transient_queue_size</code>	The transient event message queue size.
<code>event_registration_mode</code>	Event registration mode.
<code>locking</code>	Turn short locking on or off.
<code>logging</code>	Turn logging on or off.
<code>startup_script</code>	Specify executable name that will be run at database startup.
<code>versant_be_dbalogginglevel</code>	Set DBA logging level as 0 or 1 for a database.
<code>versant_be_dbalogginglevel</code>	File used to log DBA utility activity.
<code>versant_be_logfile</code>	Specify location of the per-database LOGFILE.
<code>be_syslog_level</code>	Control system logging



---

## async\_buffer\_cleaner

`async_buffer_cleaner number`

If this parameter is set to an integer greater than zero, a separate thread will be used to asynchronously flush the database server cache to disk.

The default is one.

If this parameter is set to zero, the database server process itself will flush the database server cache (no separate thread will be started.)

## async\_logger

`async_logger number`

If this parameter is set to an integer greater than zero, a separate thread will be used to asynchronously flush the database logging cache to disk.

The default is one.

If this parameter is set to zero, the database server process itself will flush the database logging cache (no separate thread will be started.)

## auto\_addvol

`auto_addvol volname_prefix size [ absolute_path | relative_path ]`

This parameter enables the automatic addvol feature. It also allows you to specify the prefix string for the volume name, locations of the new volumes as well as their size.

### Parameters are:

#### **volname\_prefix**

The prefix string for volume names. The complete volume name is generated dynamically by appending a unique constant (typically the new volume ID) to `volname_prefix`.

#### **size**

The size of data volume. Indicate kilobytes with k or K and megabytes with m or M.

For example:

1M

1024k

2000000

**For more information, See also “Size Limits Of Various Configurable Parameters” on page 117.**

**NOTE:-** All new volumes that are added will be of this size. If you want to change the volume sizes of further new volumes that will get added, then you have to modify the profile and restart the database.

### **volpath**

Actual location/directory of data volume. All the new volumes will be placed here. Versant does not check if this location has sufficient space before adding the new volume. It is the responsibility of the DBA to make sure that this path has sufficient disk space to hold the volumes.

### **NOTE:-**

- If this path needs to be changed, then you have to modify the profile and restart the database.
- On UNIX, the `volpath` directory should be owned by the DBA.
- The automatic add volume feature is currently not supported on databases that require the DBA to provide a password for access.

## **auto\_addvol\_threshold**

```
auto_addvol_threshold <absolute free space in KB/MB | percentage  
free space>  
e.g., auto_addvol_threshold 1024M OR  
      auto_addvol_threshold 10%
```

This parameter specifies when an add volume should be triggered by the server. This parameter takes value in "absolute size" or in "percentage" which would mean, trigger an `addvol` when Database has "absolute\_size" space free or Database has "percentage" space free respectively.

---

In absolute size, units of KBs or MBs like 1024M are accepted, and in percentage, values like 10%, 30% are accepted.

This is an optional parameter and the default value is 10%.

For example:

If you want to add a volume automatically when the database has 1024 Megabytes space free, you would specify 1024M as the value of this parameter.

Or If you want add a volume automatically when the database is 30% free, you would specify "30%" as the value of this parameter. A percentage would have to be specified with a "%" sign, else a parse error will be thrown.

Some valid usages are: 10% , 900K, 1024M

**NOTE:-**The old usage of specifying a "number" that meant "percentage full" is now obsolete. All existing profiles with this number will have to be updated with a value in "absolute size" or in "percentage", both meaning "free space" and not "full space".

## auto\_addvol\_aggression

`auto_addvol_aggression number`

The database server periodically checks the overall used space in the database volumes and then decides whether to add a volume or not.

Free space calculation involves examination of the existing data volumes and some computation that can take some extra CPU cycles. If your data volumes get filled up gradually over time then the server need not check the used space very frequently. This parameter allows you to specify the time interval (in seconds) after which the server checks for database used space.

Default value for this parameter is 60. This should suffice for most databases.

## Optimal Automatic Add Volume Settings

If your clients still get the Versant error 1083 - `SM_E_OUT_OF_VOL_SPACE` despite having enabled "automatic add volume" then you may need to either decrease the value of "auto\_addvol\_aggression" or increase the value of "auto\_addvol\_threshold" or both. The optimal setting really depends on the size of your database and the type of transactions that generally run on your database.

If you face problems with the `automatic addvolume` feature or need any help to tune your database settings please contact Versant Support.

### **commit\_delete**

`commit_delete ON / OFF`

Specify whether the physical deletion of objects should be delayed till commit.

The default is `OFF`.

This parameter is applicable to normal objects, i.e. non-schema objects only.

If `commit_delete` is enabled, a delete operation on an object will set the status of the object in the database as "Marked for deletion". The object will be physically deleted at commit. In case of a rollback, the object will be unmarked and the original status is restored.

If `commit_delete` is disabled, the physical deletion of the object is done immediately when the operation is sent to the database.

`commit_delete` will be effective only if logging is turned `ON`. If logging is turned `OFF`, immediate deletion is performed irrespective of the value set for `commit_delete`.

### **commit\_flush**

`commit_flush ON / OFF`

If set to `ON`, server process buffers are flushed to disk after each commit.

The default is `OFF`.

### **custom\_be\_plugins**

The implementation of Virtual Attribute Templates (VAT) are packaged as shared objects. The backend profile parameter `custom_be_plugins` is used to specify the VAT plug-ins that the server loads at system initialization. An application must specify the VAT plug-ins for the Virtual Attributes that it would like to use, before starting the database.

The syntax of `profile.be` entry to load a VAT plug-in is:

---

on UNIX:

```
custom_be_plugins <VERSANT_RUN_ROOT>/lib/lib<plugin-name>.so$  
<VERSANT_RUN_ROOT>
```

on Windows:

```
custom_be_plugins <VERSANT_RUN_ROOT>/bin/<plugin-name>.dll$  
<VERSANT_RUN_ROOT>
```

where, <VERSANT\_RUN\_ROOT> is path to the bin and lib folders in your Versant installation and <plugin-name> is one of the following:

```
tuple  
nocase  
national
```

If you want to load more than one plug-in, separate them by comma without space.

Example:

In order to load the nocase and the national plug-ins on UNIX specify:

```
custom_be_plugins <VERSANT_RUN_ROOT>/lib/libnocase.so$  
<VERSANT_RUN_ROOT>,<VERSANT_RUN_ROOT>/lib/libnational.so$  
<VERSANT_RUN_ROOT>
```

## datavol

```
datavol volume_name size [absolute_path | relative_path ]
```

The location of a particular data volume.

**Parameters are:**

**volume\_name**

The name of the data volume. The length of the name is restricted to 31 characters.

**size**

The size of the data volume. Indicate kilobytes with k or K and megabytes with m or M.

For example:

1M

1024k

2000000

**See also “Size Limits Of Various Configurable Parameters” on page 117.**

### **`absolute_path`**

The actual location of the data volume.

**CAUTION:-** If an absolute path is specified then please make sure that those paths are not referred by any other database's server process profile file (`profile.be`) that is on the same machine on which the target database directory will be created. Failure to check this can result in severe database corruption and / or deleted volumes.

### **`relative_path`**

The path to the data volume, relative to the database directory.

This parameter is relevant only to data volumes created with the `addvol` utility and not to the system, physical log, or logical log volumes.

If before creating the database, you create one or many `datavol` entries for data volumes, and then if you use `makedb` to create a server profile file, the running `createdb` will create the storage volumes as well as the system, physical log, and logical log volumes.

If you run `createdb` with the `-i` option, the volumes will also be initialized.

**For more information refer to “createdb” on page 142, in “Chapter 4 - Database Utilities”.**

If you create an entry for a data volume that does exist, then whenever the database starts, the system will look for the data volume at the specified location.

**For more information refer to “Database Creation Basics” on page 54, in “Chapter 2 - Database Creation”.**

---

## locking

`locking ON|OFF`

The `locking` parameter specifies whether short locking is `ON` or `OFF`:

If `locking` is `OFF`, setting of short locks is disabled. Set `locking` to `ON` to enable the setting of short locks.

The default for personal databases is `locking OFF`. The default for group databases is `locking ON`.

Locking controls concurrent access to objects by different processes, and it provides data integrity at the price of performance.

It is safe to turn short locking `OFF` in a personal database, if you are running only one application process. If multiple applications are accessing a group database, locking should always be `ON` to ensure orderly concurrent access to objects.

## logging

`logging ON|OFF`

Specify whether logical and physical logging are enabled.

The default for both personal and group databases is `ON`.

Logging guarantees that a database can at least be restarted and accessed after a system failure. It also keeps track of the status of transactions and savepoints, and logging ensures database consistency. All recovery processes can recover from crashes that occur during recovery.

Logging must be turned `ON` to perform transaction rollbacks and/or to use savepoints.

Using a raw device for the logical and physical log volume improves performance when logging is enabled.

Logging overhead can be substantial, which is why Versant implements single user personal databases where logging can be turned `OFF`. However, turning logging `OFF` means that full recovery from a system crash may not be possible even though no data is lost.

All transaction commits are two-phase, which ensures the consistency of a committed transaction across a distributed database system.

### Logging Usage Notes

- Logging should always be `ON` in group databases.

Although you can safely turn `OFF` logging in a personal database to reduce overhead and improve performance, we strongly recommend that you keep logging `ON` in group databases.

- In group databases, do not use logging without locking.

## startup\_script

```
startup_script script_file_name [output_logfile_name]
```

The `startup_script` parameter specifies the executable that should be launched at database startup. The startup executable can either be a binary or a script.

### Parameters are:

#### `script_file_name`

Name of the executable. It is relative to the database directory. Specifying an absolute path is not allowed. This executable will be launched at the time of the database startup.

For security reasons, only the DBA should have write permissions to modify this file.

On Windows, the extensions `".bat"` and `".exe"` are the only ones considered to be valid executables. On UNIX, there is no such restriction.

#### `output_logfile_name`

An optional path to the output log file. An absolute (actual location) or relative (to the database directory) path can be used.

The startup executable starts running at database startup. The database server redirects the `STDOUT` and `STDERR` of the startup executable to this log file. The output is always appended to the startup executable log file.

If `output_logfile_name` is not specified, then it will be created in the database directory and the name of this log file will be obtained by appending the `script_file_name` with `".log"` extension.

For example if `script_file_name` is `startupscrip.sh`, then the output of the script will be saved in the database directory in a file named `startupscrip.sh.log`.

There could be problems executing the startup script due to the access or execute permissions. In such cases the database startup will return with appropriate errors. After the startup executable has been launched successfully, the DBA may need to periodically check the startup executable log file for the error status, if any.



## Monitoring startup executable

To know whether the startup executable is in execution, get the process-id in the following ways:

### 1. Check the output of `dbtool -sys -info -resource <dbname>`

The output of `dbtool -sys -info -resource` gives the process-id of the startup executable if it is executing.

Sample output would be:

Processes and Threads:

PID	TID	ConnID	State	Type
5898	1	2	alive	Cleanbe-proc
5898	2	3	alive	Logger/Flusher
5898	3	4	alive	Logger/Flusher
5899	1	5	alive	Obe-proc
5899	2	6	alive	Cleanbe-thrd
<b>5906</b>			<b>alive</b>	<b>Startup executable</b>

### 2. Reading from LOGFILE

Whenever the startup executable is launched the process-id is written to the database LOGFILE.

Sample output would be:

```
PID: 5812; HOST: everest; Tue May  2 02:27:35 2006
INVOKED; /.../db/t.sh; CWD: /.../db; RUID: yphadke; EUID: yphadke;
Process:5812  Tue May  2 02:27:35 2006
Startup executable running with process-id 5906
```

The process-id in the LOGFILE would be useful to find and track the startup executable (if required).

### NOTE:-

1. `stopdb` will not terminate the startup executable. A message "Startup executable running after database down. process-id 12103" will appear in the database LOGFILE stating that the startup executable is not terminated.
2. If database startup detects that previous startup executable is active, then the database startup will fail.

## startup\_script Usage Notes

- It should not use any system call or command that could create a new process.

- It should not use `removedb / stopdb` or any other commands that stop the database.
- The following environment variables are available for the user within the startup executables:
  - `VERSANT_STARTUP_DBNAME`: The name of the database. This could be used for executing any versant utilities that require the database name as parameter.
  - `VERSANT_STARTUP_LOG`: The name and path of the startup executable log file.

**WARNING:-** Starting any background processes should be avoided. If a background process is started, then the user should keep track of that process(s).

### Example

The following is a sample Bourne shell script that can be used as a startup executable to take a level 0 backup and start roll forward archiving:

```
#!/bin/sh
# take a level 0 backup and start roll-forward logging
vbackup -device $VERSANT_STARTUP_DBNAME.backup -rollforward -backup
$ VERSANT_STARTUP_DBNAME

if [ $? -ne 0 ]
then
    echo "level 0 backup failed"
    exit 1
fi

# start the roll-forward archiver and create files of size 2MB with
auto
filename generation
vbackup -device $ VERSANT_STARTUP_DBNAME.roll -noprompt -capacity 2M
-log $ VERSANT_STARTUP_DBNAME

if [ $? -ne 0 ]
then
    echo "roll forward logging failed"
    exit 1
fi

exit 0
```

---

## versant\_be\_dbalogginglevel

This is the DBA utility logging level.

You can set the value of this parameter as 0 or 1.

The value 1 implies DBA logging on and 0 implies DBA logging off.

To enable/disable logging, edit the database profile file `profile.be`, and set the `versant_be_dbalogginglevel` parameter to an appropriate value.

By default, the DBA logging will occur (i.e. value is 1).

## versant\_be\_dbalogginglevel

Specify the name of the file that would be used to log DBA utility activity.

There is no default value set for this parameter. In the absence of this parameter, the per-database configurable `LOGFILE` in the individual database directory will be used. This file will be created if it does not exist. If it already exists, then file will not be overwritten. Logging messages will be appended to the end of the file.

For UNIX, relevant OS permissions are always required to create or open a file.

**For more information refer to “LOGFILE” on page 297, in "Chapter 5 - Directories and Files".**

## versant\_be\_logfile

Specify the location of the per-database configurable `LOGFILE`. All information (database and system and error messages) is logged here.

The default value is the `LOGFILE` in the database directory under the database root directory. If specified in the `profile.be`, this file name will be used.

## be\_syslog\_level

This parameter controls system logging for critical database operations.

The value of this parameter can currently be set to 0 or 1.

Value 1 implies perform logging, while 0 implies no logging.

The logging is directed to system logger on unix(i.e. `syslog()`) and to the application event log on Windows. The events that are logged are starting and stopping of a database and crashes of the obe process.

## User-Authentication using Third Party plug-ins

A third party authentication system is used for user authentication, by using `custom_be_plugins`.

In this case, the user profile along with the password will be managed directly by the third party system, for example LDAP. The authentication is done using a plugin provided by the user.

The DBA must indicate the location of the plug-in in the `profile.be` file.

**For implementing the user authentication plug-in, please refer to the chapter “Controlling Access To Database” in *Versant Database Fundamentals Manual*.**

## Tuning Parameters

**Tuning parameters:** Versant uses these each time you start the database. You can edit these parameters to change the operating parameters of your database.

Most of these parameters are optional, as the default values are appropriate for most situations. However, the parameter `class` is crucial to databases containing a large number of classes - if more than the default number of 480 classes will be accessed during the time a database is running, you will need to specify a new value for `class`.

Parameter	Description
<code>bf_dirty_high_water_mark</code>	When the cleaner thread starts flushing dirty pages.
<code>bf_dirty_low_water_mark</code>	When the cleanup thread stops flushing dirty pages.
<code>class</code>	The number of cached user defined classes.
<code>db_timeout</code>	Database timeout.
<code>heap_size</code>	A hint for the initial size of server heap.
<code>heap_size_increment</code>	Size by which server heap can be expanded.

---

index	The number of cached user defined indexes.
llog_buf_size	Size of the logical logging buffer in bytes.
lock_batch_size	Number of lock requests to be batched together
lock_wait_timeout	Time to wait for a lock.
max_page_bufs	The number cache buffers.
multi_latch	Database latching behavior.
obe_port_begin	Specify starting port number for connections. This parameter is no longer supported by VERSANT, please use the parameter VERSANT_SERVER_PORTS instead.
	<b>For more details refer to “VERSANT_SERVER_PORTS” on page 346 in "Chapter 6 - Configuration Parameters".</b>
obe_port_end	Specify ending port number for connections. This parameter is no longer supported by VERSANT, please use the parameter VERSANT_SERVER_PORTS instead.
	<b>For more details refer to “VERSANT_SERVER_PORTS” on page 346 in "Chapter 6 - Configuration Parameters".</b>
plog_buf_size	Size of the physical logging buffer in bytes.
polling_optimize	Strategy for FTS recovery.
spin_count	Spin lock busy wait loop count
transaction	The maximum number of concurrent transactions.
user	Initial size of user names list.
volume	The number of cached data volumes.
logvolmaxsize	The maximum size to which the logical and physical log volumes can grow

Following are the Tuning parameters:

## bf\_dirty\_high\_water\_mark

`bf_dirty_high_water_mark` *number*

Specify, in number of pages, when an asynchronous page cleaner thread will start to flush to disk dirty pages in the buffer pool.

The default value of `bf_dirty_high_water_mark` is 512 pages.

Together, the high water mark and lower water mark parameters influence when the asynchronous buffer cleaner writes dirty buffer pages to disk.

Without the lower water mark concept, pages that are repeatedly set to dirty are flushed over and over again. With no gap between the high and low water marks, a page that is repeatedly set to dirty does not get flushed in time. The range in between ensures proper flushing.

In general, pages are flushed in priority order, and pages with a numerically lower priority are flushed first. In general, root and branch pages of an index have priority 3, leaf pages of an index have priority 2, and data pages have priority 1. A time algorithm will decrement less used pages to priority 0. However, when you dirty a page, its priority is set back to its correct priority number (for example, a data page will be set to priority 1.)

The asynchronous buffer cleaner wakes up whenever the number of dirty pages in the buffer cache reaches the high water mark. When the number of dirty pages in the buffer cache reaches the high water mark, dirty pages of all priority are flushed until the number of dirty pages reaches the low water mark. The buffer cleaner then flushes all remaining dirty pages of priority 0 and then goes to sleep. Without the low water mark concept, pages that are repeatedly dirtied would otherwise get flushed over and over again, which would degrade performance. Without there being a gap between the high and low water marks, a page that is repeatedly dirtied might go for a long time without being flushed.

For example, suppose that the high water mark is 512 pages and the lower water mark is 204 pages. When the number of dirty pages reaches 512, the buffer cleaner writes dirty pages of any priority until there are only 204 dirty pages left in the buffer cache. From that point on, only pages of priority 0 are flushed.

## bf\_dirty\_low\_water\_mark

`bf_dirty_low_water_mark` *number*

Specify, in number of pages, when an asynchronous cleanup process or thread will stop clearing the buffer pool of pages containing dirty objects.

---

The default value of `bf_dirty_low_water_mark` is 204 pages.

**See also “Tuning Parameters” on page 106.**

## class

`class number`

Specify the integer number of cached user defined classes.

If not specified, the default number of cached user defined classes is 480.

If more than the specified or default number of classes are accessed during the time a database is running, you will get an error.

## db\_timeout

`db_timeout delay`

Specify that the database is to be automatically shut down if there have been no active transactions for `delay` number of minutes.

The default delay is -1, which means no time-out. All negative values also mean no time-out.

For example, to specify automatic shutdown after one hour:

`db_timeout 60`

## heap\_multi\_region

`heap_multi_region ON / OFF`

This parameter is used to enable/disable the “Multi region heap” feature.

The default value is `ON`.

This parameter is available on Windows operating systems only. On UNIX operating systems, use of this parameter is invalid and will be considered as an error.

The “Multi region heap” feature allows Versant to split its server heap across multiple regions in the virtual address space of the process.

This has the following advantages:

- It avoids memory conflicts with DLL's that get loaded into the virtual address space of the server process
- It allows for the creation of a much larger heap than what was possible earlier
- The “Multi region heap” feature is scalable when 3GB large memory addressing is configured on the system.

**For more details on how to do 3GB configuration, please refer to Microsoft Help and Support.**

## heap\_size (database)

`heap_size size`

A pre-allocation hint for the initial size of the server process heap in shared memory.

Indicate kilobytes with a `K`. If you do not specify a value, Versant will determine a value.

The server process heap is memory from which internal data structures and server process buffers are allocated, and its size sets a limit on the maximum size of regular object instances.

If you know that a server process will require a lot of memory, you can increase this parameter in order to preallocate space.

If more memory than the initial amount specified by this parameter is needed, the heap size will be dynamically increased as needed. If `heap_size` is too large, you may not be able to allocate shared memory when the database is started.

**See also “heap\_size (application)” on page 80, of application process parameter.**

## heap\_size\_increment (database)

`heap_size_increment size`

Set the size by which the server heap will be expanded as needed.

The default increment is 2 megabytes.

**See also “heap\_size (database)” on page 110, of server process parameter.**



---

## index

`index number`

Specify the integer number of cached user defined indexes.

If not specified, the default number of cached user defined indexes is 480.

If more than the specified or default number of indexes are accessed during the time a database is running, you will get an error.

## logvolmaxsize

`logvolmaxsize`

Specify the maximum amount of disk space that would be available for the logical and physical log volume to grow.

This parameter will remain effective until the database is shutdown. To change the value of the parameter DBA should shutdown the database and then restart it.

Consider the following scenarios:

**Scenario 1:** The size of the logical log file at database startup time is already greater than the value specified for `logvolmaxsize` in `profile.be`. In such a case the size of the logical log file at startup time (which is larger than the specified `logvolmaxsize` parameter) will be used as the max size, and the value specified in the `logvolmaxsize` parameter will be ignored.

**Scenario 2:** If the value of `logvolmaxsize` is less than the size of the llog or plog volume (as specified in the `llogvol/plogvol` parameters) then the value of `logvolmaxsize` will implicitly get increased to the larger of `llogvol` and `plogvol` sizes.

**Scenario 3:** If a value of 0 (or a negative value) is given for the `logvolmaxsize` parameter, then it is equivalent to not specifying this parameter at all in `profile.be`.

If the DBA has set the `logvolmaxsize` parameter to a value that is more than the actual space available on disk, then the server may still crash if the disk itself runs out of space.

## llog\_buf\_size

`llog_buf_size size`

Specify the logical log buffer size in bytes.

The default is: `llog_buf_size 161k`

The larger the logical log buffer, the less frequently the system needs to flush the buffer to disk. Decreasing this parameter will save some memory but increase the frequency of disk writes. Increasing this parameter will slightly reduce the frequency of disk writes but may waste memory and cause paging. The default of 161K should work well for most cases.

The logical log buffer is dynamically divided into two units. This allows use of a double buffering strategy to allow parallel execution of disk input/output and CPU execution.

## lock\_batch\_size

`lock_batch_size number`

This parameter signifies the number of lock requests to be batched together and processed under one acquisition of the lock latch during acquire/backout locks. Batching the lock latch requests together reduces the lock latch contention thereby, reducing the overhead of context switches and therefore improves performance.

The default value of this parameter is 200.

If your applications are locking more than 200 objects in a transaction then you may want to increase this value to get optimum performance.

## lock\_wait\_timeout

`lock_wait_timeout number_of_seconds`

The number of seconds the server waits for a lock on an object to be released before generating an `object locked` error.

The default is to wait for 60 seconds: `lock_wait_timeout 60`

Use the value -1 to specify waiting forever and use 0 to specify immediate return.

If a lock cannot be acquired in the specified number of seconds, the following error message will be returned:

```
2903 SM_LOCK_TIMEOUT Lock wait timed out.
```

---

## max\_page\_buffs

`max_page_buffs number_pages`

Specify the maximum number of 16K buffers for caching disk pages from data volumes.

This parameter strongly influences system performance. If you set it too low, the disk input/output increases. If you set it too high, the physical memory becomes exhausted and virtual memory swapping to disk takes place, which defeats the purpose of caching disk pages. If you have a large database you can sometimes increase performance by increasing this parameter. If you have a small database, you may want to decrease it to save memory.

The default value of 1024 should work well for most applications.

## max\_sorting\_memory\_per\_query

`max_sorting_memory_per_query size`

This parameter indicates the maximum size to which the sorting memory for each query can grow. The amount of memory can be specified in kilobytes (K) or megabytes (M).

If `max_sorting_memory_per_query` is specified, then there will be no limit for total memory used for sorting. But individual query cannot use more than the size specified for sorting memory.

If the memory used for sorting a particular query exceeds the size of memory specified by `max_sorting_memory_per_query` in `profile.be`, the error

`QRY_SORT_MEMORY_EXCEEDED_QUERY_MAX` will be returned and the query will be aborted.

The following example limits sorting memory per query to 5M:

```
# The maximum size to which the sorting memory for each
  thread can grow
  max_sorting_memory_per_query 5000K
```

## max\_sorting\_memory\_total

`max_sorting_memory_total size`

This parameter indicates the maximum size to which the sorting memory areas of all on-going queries put together can grow. The amount of memory can be specified in kilobytes (K) or megabytes (M).

If the `max_sorting_memory_total` is specified, there will be no limit for individual query. But the total memory consumed for sorting cannot exceed the size specified.

If the total memory used for sorting exceeds the size specified by parameter `max_sorting_memory_total` in `profile.be`, the query that is trying to allocate memory for sorting will be aborted and the error `QRY_SORT_MEMORY_EXCEEDED_TOTAL_MAX` will be returned.

If `max_sorting_memory_total` is smaller than `max_sorting_memory_per_query`, then `max_sorting_memory_per_query` will be ignored.

The following example limits the total memory used for sorting to 10M:

```
# The maximum size to which the sorting memory areas of
  all threads put together can grow
max_sorting_memory_total    10M
```

## multi\_latch

`multi_latch` *ON/OFF*

If the `multi_latch` parameter is set to `ON`, the database server process will use multiple latches. Each latch will protect a different sub-module. This is the default.

If the `multi_latch` parameter is set to `OFF`, the database server process will use a single latch to protect its shared memory areas.

For a group database, the `ON` default improves concurrency when multiple users are accessing the database on a multiple processor machine. For single processor machines, setting `multi_latch` to `ON` may not improve your performance. For a personal database, setting this parameter to `OFF` improves performance when the database is accessed by a single user.

## obe\_port\_begin

This parameter is no longer supported by VERSANT please use the parameter `VERSANT_SERVER_PORTS` instead.

**For more details on `VERSANT_SERVER_PORTS`, please refer to the section on “`VERSANT_SERVER_PORTS`” on page 346 in “Chapter 6 - Configuration Parameters”.**

## obe\_port\_end

This parameter is no longer supported by VERSANT please use the parameter `VERSANT_SERVER_PORTS` instead.

---

For more details on `VERSANT_SERVER_PORTS`, please refer to the section on “`VERSANT_SERVER_PORTS`” on page 346 in “Chapter 6 - Configuration Parameters”.

## **plog\_buf\_size**

`plog_buf_size` *size*

Specify the physical log buffer size in bytes.

The default is: `plog_buf_size 161k`

The larger the physical log buffer, the less frequently the system needs to flush the buffer to disk. Decreasing this parameter will save some memory but increase the frequency of disk writes. Increasing this parameter will slightly reduce the frequency of disk writes but may waste memory and cause paging. The default of `161k` should work well for most cases.

The physical log buffer is dynamically divided into two units. This allows use of a double buffering strategy to allow parallel execution of disk input/output and CPU execution.

## **polling\_optimize**

`polling_optimize` *ON|OFF*

Set Fault Tolerant Server recovery strategy.

### **Options are:**

#### **OFF**

The default is `OFF`, which prevents applications from getting polling induced deadlock errors.

If `polling_optimize` is `OFF`, the polling process, while resynchronizing the replica pair after a failure, applies changes one object at a time. Each object is first locked, then the update is applied, and then the lock is released. This prevents polling induced deadlock.

#### **ON**

The default technique of applying changes one at a time is inefficient.

If `polling_optimize` is `ON`, objects are group read, then changed, then released. This greatly improves performance. However, this might cause a polling induced deadlock error. If `polling_optimize` is `ON`, applications should be able to handle deadlock errors.

## spin\_count

`spin_count number`

The VOD server uses spin locks to protect access to certain critical sections. The spin lock is implemented using a busy wait loop for a certain number of iterations, after which the threads sleep for a while before trying again. The ideal value for these iterations or “count” depends on the number of requests for the critical section/thread scheduling/processor architecture.

The spin count parameter allows you to adjust this count to the best suited value for your server. The default value is 2000 and the minimum allowed value is 1. Keeping the value too low will put the contending threads to sleep too soon and keeping it too high might waste unnecessary CPU cycles.

Determining the optimal spin count for your server can be complicated. For this purpose a new statistic “db\_spinlock\_misses” has been provided. This statistic gets updated everytime a thread goes to sleep after a busy wait loop in the spin lock. Ideally this statistic should be zero but if you see it increasing on your production system you might want to increase the spin count and remonitor this statistic. This will help you achieve the ideal spin count.

## transaction

`transaction number`

The maximum number of concurrent transactions allowed for this database.

The default value is 10.

If more than the specified or default number of transactions are used during the time a database is running, you will get an error.

## user

`user number`

Specify the initial size of named users list and used as hint during startup phase. If there are more users then the size will be automatically adjusted.

This user list can be viewed with `dbuser -list`. For very small databases this parameter might be reduced. If not specified, the default initial size is 8.

---

## volume

*volume number*

Specify the integer number of cached data volumes.

If not specified, the default number of cached data volumes is 16.

If more than the specified or default number of data volumes are used during the time a database is running, you will get an error.

## Size Limits Of Various Configurable Parameters

Size Limits	Min Value	Max Value	Default Value	Limit imposed by OS
Data Volumes	2M	64G (On Solaris) 4G (On Windows)	128M	large file support on Solaris 2.6 and above. Refer <a href="http://www.sun.com/software/white-papers/largefiles/largefiles.pdf">http://www.sun.com/software/white-papers/largefiles/largefiles.pdf</a> for more information. On Windows, FAT file system does not support files of size greater than 4G. NTFS provides large file support.
Server Page Cache (in no. of buffers)	10	no limit	1024	The Server Page Cache can be configured to the max. value as long as the heap could grow. See Heap Size limits for further information.

System Segment Cache (no. of entries)	class 480 index 480 user 8 volume 16	no limit	class 480 index 480 user 8 volume 16	The number of entries in system segment cache can be configured to the max. value as long as the heap could grow. See Heap Size limits for further information.
Schema Cache	-	-	-	
Log Volumes	2M	64G	2M	
Log Buffers	161K	2G	161K	The log buffer can be configured to the max. value as long as the heap could grow. See Heap Size limits for further information.
Heap	2M	2G (32-bit system) no limit (64-bit system)	2M	Virtual address space that is accessible for user-mode and kernel-mode threads in a process. This range of virtual address space is generally the remainder of the process' virtual address space that is reserved for system use.
Database Backup, Roll Forward Archive media	-	No limit (on Solaris) 4G (On Windows)	-	Large file support on Solaris 2.6 and above. Refer <a href="http://www.sun.com/software/white-papers/largefiles/largefiles.pdf">http://www.sun.com/software/white-papers/largefiles/largefiles.pdf</a> for more information. On Windows, FAT file system does not support files of size greater than 4G. NTFS provides large file support.



---

## Statistics Parameters

**Statistical parameters:** Versant can collect statistics and vary assertion levels.

Parameter	Description
<code>stat</code>	Turn collection of connection and database statistics on or off.
<code>assertion_level</code>	Type of assertions.

Following are the Statistical parameters:

### **assertion\_level**

`assertion_level` *number*

Type of assertions.

You can set this parameter to levels between 0 and 2.

If you set it to 0, the system performs only simple assertions. The higher the level, the more expensive the checking.

The default is 0.

### **stat**

`stat [ stat_name ] action`

Turn collection of connection and database statistics on or off for this database.

Alternatives for `action` are:

**on**

Collect statistics each time an application connects to this database and continue unless explicitly stopped with `vstats` or `o_collectstats()`.

**off**

Stop collecting statistics.

By default, statistics collection is `OFF`.

Alternatives for the optional `stat_name` parameter are:

### **all**

If you specify the special value `all`, collection of all database and connection statistics will be turned `OFF` or `ON` per the `action` parameter. This is the same as supplying no value for `stat_name`. This is default option.

### **be\_all**

If you specify the special value `be_all`, collection of all connection statistics will be turned `OFF` or `ON` per the `action` parameter.

### **db\_all**

If you specify the special value `be_all`, collection of all database statistics will be turned `OFF` or `ON` per the `action` parameter.

## **Connection or database statistic**

If the name of a connection or database statistic is specified, collection of the specified statistic will be turned `OFF` or `ON` per the `action` parameter.

The names of connection statistics have the prefix `be_`. The names of database statistics have the prefix `db_`.

**For a list of connection and database statistic names, see the reference for the `vstats` utility in the *Versant Database Administration Manual*.**

Statistic names must be specified without the `STAT_` prefix and transposed to lower case.

You can create multiple entries for `stat` in the server process profile to specify the various statistics that you want.

Collection of database statistics begins when the database is started. Collection of connection statistics begins with the database connection.

Statistics are stored in memory where they may be viewed with the `vstats` utility or brought into an application with the `C/Versant o_collectstats()` function.

You cannot use `stat` to collect function, application, or session statistics.

Even when database replication has been turned `ON`, statistics collection tools and mechanisms operate only on named databases and not on replica databases. To collect statistics for a replica database, you must apply statistics collection mechanisms specifically to the replica database.

**For more information on statistics collection and usage notes, please refer to the *Versant Database Fundamentals Manual*.**

## Tracing Parameters

**Tracing parameters:** Versant has tracing facilities. You can set the number of traces and the file in which the system stores the traces.

Parameter	Description
<code>trace_comps</code>	Database components to be traced.
<code>trace_entries</code>	Number of trace entries to be maintained in the trace file.
<code>trace_file</code>	Name of the trace file.

Following are the tracing parameters:

### `trace_comps`

`trace_comps options`

Database server components to be traced.

The `options` parameter can contain one or all of the following elements. If multiple elements are specified, the list must be delimited by commas with no spaces.

Elements	Description	Elements	Description
<code>am</code>	Trace the current state of queries or cursors.	<code>tr</code>	Trace transaction operations.

---

at	Trace entries created, updated, and deleted in the internal AT table.	net	Trace the network layer
bf	Trace the buffer manager.	sl	Trace the system layer
bt	Trace the B-tree manager	vs1	Trace the Virtual System layer
dr	Trace the file directory	ut	Trace utilities
hs	Trace the Hash-Indexing module	om	Trace the object manager
io	Trace the I/O module	qry	Trace the query APIs
lg	Trace logging and recovery.	lk	Trace the lock manager
ps	Trace process/thread operations.	rcv	Trace the recovery manager
rc	Trace object creation, modification, and deletion.	csr	Trace the cursor APIs
sd	Trace the system segments	ev	Trace the Event Notification
si	Trace the storage manager interface	api	Trace the Kernel APIs
sm	Trace the storage manager	rpc	Trace RPCs

## trace\_entries

`trace_entries` *number*

Number of trace entries to be maintained in the trace file.

The default is 10000 entries, after which entries are reused.

## trace\_file

`trace_file` *name*

Name of the trace file.

By default, the trace file will be created in the database directory.

---

On UNIX, the default name of the trace file is `.systrace`.

On Windows, the default name of the trace file is `systrace`.

If the entry begins with a forward "/" (UNIX) or a backwards slash "\" (PCs,) the `name` parameter is treated as the full path, including the file name. Otherwise, the `name` parameter is assumed to be just the file name, and the file will be created in the database directory.

**For more information refer to “dbtool” on page 155, in "Chapter 4 - Database Utilities".**

## Authentication Parameter

**Authentication parameter:** Versant has authentication parameters used for user authentication.

Parameter	Description
<code>Authentication_program</code>	Set location of user authentication program.
<code>be_permit_pre70_clients</code>	Permit Pre7.0 clients to connect

### authentication\_program

`authentication_program program_path`

To specify the location of a customized user authentication program, on the database machine, in the database profile file, add a line containing the keyword `authentication_program` followed by the path to the database authentication program.

### be\_permit\_pre70\_clients

`be_permit_pre70_clients OFF/ON`

Permit Pre 7.0 clients to connect.

Default is `OFF`, which prevents the pre 7.0 clients to connect to database.

If `be_permit_pre70_clients` is `ON`, the pre 7.0 clients can connect.

---

**NOTE:-** Even if this parameter is set, pre 7.0-DBA specific utility connections will not be permitted.

## Event Notification Parameters

**Event notification parameters:** Versant can save event registrations and generated events.

Parameter	Description
<code>event_registration_mode</code>	Persistent or transient registrations.
<code>event_msg_mode</code>	Persistent or transient messages.
<code>event_daemon_notification</code>	Notification type when event daemon goes down.

### event\_daemon

`event_daemon code_path [parameters]`

The full path name and optional parameters to code that will start an event delivery daemon.

You must separately write this code, which is operating system specific. This code must be on the same machine as the database.

When a database invokes the executable at `code_path`, it inserts the database name as the first parameter and then follows it with your parameters. If you wish to use multiple parameters, you must enclose them in double quotes and then parse the parameters yourself.

**For more information, please refer to the chapter "Versant Event Notification" in the *Versant Database Fundamentals Manual*.**

`event_daemon MANUAL`

The default entry, meaning that no event daemon will be started when the database starts. This means that you must manually start an event delivery daemon after the database starts.

---

## event\_msg\_mode

```
event_msg_mode { persistent | transient }
```

Specify the persistence of generated events when using event notification features.

Event notification modes are:

### **persistent**

Store generated events in the database.

### **transient**

Keep generated events in the memory. This is the default mode.

## event\_msg\_transient\_queue\_size

```
event_msg_transient_queue_size size
```

The maximum number of per-transaction event messages that can be held in the transient event queue.

The default transient event message queue size is 20480 bytes. You can set it to any value between 0 and  $2^{32}$ , non-inclusive.

The persistent event message queue size is  $2^{32}$ , and it cannot be changed.

Versant does not use the Operating System message queue.

In UNIX, the event message size is unlimited.

## event\_registration\_mode

```
event_registration_mode { persistent | transient | old_transient }
```

Specify the persistence of registrations when using event notification features.

Event registration modes are:

### **persistent**

Store registrations in the database.

## **transient**

Keep registrations in the memory.

## **old\_transient**

Maintain backward compatibility with applications using Versant releases prior to 6.0.5.0

This is default mode.

## **event \_daemon\_notification**

`event_daemon_notification {off | on}`

Set the type of notification strategy when event daemon is down.

**off**: Default is **OFF**. Write the error to the database LOGFILE but don't abort the commit.

**on**: Abort the commit and throw the error to the client in addition to writing the error to the database LOGFILE.

**For more information, please refer to the usage notes in the Chapter "Event Notification" in *Versant Database Fundamentals Manual*.**

## **Database Space Parameter**

**Database space parameters:** Versant has parameters to know the space of the database.

Parameter	Description
<code>Out_of_space_warning_thres hold</code>	Threshold value that triggers generation of system messages
<code>out_of_space_warning_thres hold_increment</code>	Indicates the increment at which a warning message should be generated

## **out\_of\_space\_warning\_threshold**

`out_of_space_warning_threshold   percent`



---

This threshold triggers the generation of system messages when the database volume space for a specific class reaches this threshold limit. The default value for this parameter is 90% i.e. if the database volume space is 90% full. Thereafter the system will log system messages into the file at every user-specified increment until the database volume space is exhausted. The space is measured in terms of extents.

## **out\_of\_space\_warning\_threshold\_increment**

`out_of_space_warning_threshold_increment`   percentage

This parameter indicates the increment at which a warning message should be logged into the per-database configurable `LOGFILE`. This parameter is defined in terms of percentage and has a valid range from 1 to 100. The space is measured in terms of extents.

**For more information refer to “LOGFILE” on page 297, in "Chapter 5 - Directories and Files".**



---

This Chapter gives us detailed explanation about various Versant Database Utilities.

Some utilities can be used remotely and hence wherever their names differ by operating system (Unix/PC), they are shown separately.

This Chapter describes the following:

- Quick Reference
- Utility Reference

## QUICK REFERENCE

Sr. No	Utility Name	Function
1.	"addvol"	To increase the storage capacity of the database by creating and adding a volume to the database
2.	"cleanbe"	To clean up unused resources at the backend.
3.	"cleanfe"	To clean up unused resources at the frontend.
4.	"comparedb"	To count the objects in the databases and confirm that the LOIDs for the objects are same. Also displays the LOID for the first object that is different.
5.	"cnvrtdb"	To convert an existing database created with a prior Versant release to work with the current Versant release.
6.	"createdb"	To create, format and initialize a new database.
7.	"creatrep"	To create a replica database and then load the replica database with the contents of a primary database.
8.	"db2tty"	To display the contents of the database specified along with its classes and objects.
9.	"dbid"	To manage the osc-dbid file.
10.	"dbinfo"	To determine or set the current mode of the database.

- 
- |     |                   |  |
|-----|-------------------|--|
| 11. | "dblist"          | To list the databases in a system of databases coordinated by an <code>osc-dbid</code> file.   |
| 12. | "dbtool"          | Depending upon options specified, you can use the this utility to get information about locks, objects, transactions, classes, logging activities, database system, storage volumes, indexes, AT table and event notification information. |
| 13. | "dbuser"          | To add, delete, list users or change password of a database or a DBA.  |
| 14. | "dropcls"         | To drop classes, subclasses and its instances from the specified database.   |
| 15. | "dropinst"        | To remove all instances of a class or classes and its subclasses from a database without changing the database schema.   |
| 16. | "ftstool"         | To control the automatic synchronization of the Fault Tolerant Server, to get the database and polling state of the FTS database and to force a database fail-over for administration purpose, whenever necessary.                         |
| 17. | "makedb"          | To create a database directory as per the specified options and to create the database support files.  |
| 18. | "makeprofile"     | To make the application process and server process profiles for a database.  |
| 19. | "NetworkServices" | To configure the Versant service, <code>versantd</code> and to listen and/or respond on multiple ports   |
| 20. | "oscp"            | To display information about the Versant environment for the currently used release.   |

- |     |            |  |
|-----|------------|--|
| 21. | "polling"  | To synchronize two databases in a replication pair, using the database specified as a source.                                      |
| 22. | "removedb" | To stop the database if it is running, destroy and remove all volumes and delete the database from the <code>osc-dbid</code> file. |
| 23. | "removrep" | To stop the synchronous database replication, remove the synchronous replication database.   |
| 24. | "reorgdb"  | To reorganize all data volumes of the database by rearranging objects to remove gaps in physical space.                            |
| 25. | "sch2db"   | To load the class definitions in the schema definition file in the database.   |
| 26. | "schcomp"  | To create a schema description file and a schema definition file, if given an implementation file.                                 |
| 27. | "setdbid"  | To set the database identifier of a specific database to a specified value.  |
| 28. | "ss.d"     | To add a new command line option to the SS daemon to invoke it from the command line.  |
| 29. | "startdb"  | To start the specified database.   |
| 30. | "stopdb"   | To stop the specified database and removes all database resources in memory.   |

---

31.	"vbackup"	To manage database backup, restore and roll forward archiving operations.
32.	"vcopydb"	To copy all objects and class definitions from one database to another.
33.	"vinfo"	To display Versant product information, list the version information for each component or verify consistency of the installation.
34.	"verr"	To print an error message given an error number, name or fragment.
35.	"verrindx"	To index the new error message created.
36.	"vinstinf"	To print the details of the installation information needed for generation of the license keys.
37.	"vlicchk"	To determine whether a specified component is licensed.
38.	"vlicvrfy"	To scan a specified license file for errors.
39.	"vmovedb"	To move all objects and copy class definitions schema, from source to target database.
40.	"vstats"	To start the statistics tool which can collect and view database statistics.
41.	"vstream"	To export the contents of a database to a file or import data from a file into a database.

**42.**    “vtape”

To view and set SCSI tape drive and media parameters.



---

## UTILITY REFERENCE

### addvol

```
addvol parameters [options] dbname
```

This utility increases the storage capacity of the database by creating and adding a volume to the database `dbname` - with the name and path specified as parameters substituted for `parameters` and with the options substituted for `options`.

For a remote database, append the node name to the database name using the syntax `database@node`.

The current limitation on the maximum number of database volumes allowed is 240.

You must be the owner of the database to add a volume.

The new volume will be registered in the database, and the server process profile file `profile.be` will contain an entry of following type indicating volume name, size and absolute location specified at creation time.

```
datavol volume_name size [absolute_path]
datavol voll          1M    /voll/voll
```

The new space will be available immediately after the `addvol` utility finishes. The additional storage space will be available to all classes in the database as classes can span volumes. The volume added would augment the system volume but have no effect on the capacity of the logical or physical log volumes.

There is no corresponding command to delete a volume. To shrink the space allocated to a database, use the `reorgdb` utility.

The minimum size of a database volume depends upon the setting for `extent_size` specified in the server process profile file `profile.be`.

**See also the `sysvol` and `extent_size` parameters.**

If you run out of database space or database recovery fails for lack of space, following error will be raised:

```
error 1083, SM_E_OUT_OF_VOL_SPACE: all volumes exhausted.
```

**Mandatory parameters to be specified are:**

**-n volName**

Name for the new volume. The length of the name is restricted to 31 characters.

For example: `-n volume2`

**-p volPath**

Full path name of the volume device or file.

You may specify the path in relative or absolute terms. You can specify either a file volume or a raw device.

For example: `-p volume2`

The data volume can be moved later to another location. The `datavol` entry in the server profile will have to be modified accordingly.

**For more information refer to “datavol” on page 99, in "Chapter 3 - Database Profiles".**

**Options are:**

The default option is:

**-e extentsize**

Extent size in pages. The default is to use the extent size specified in the server process file `profile.be`.

For example: `-e 4`

**-i**

Pre-allocate disk space and initialize the volume. This will prevent a possible insufficient space error at runtime. The `-i` parameter is ignored if you specified a raw device for the volume.

**-noprint**

Suppress display messages while running.

**-s volSize**

Volume size.

The default size is 128 megabytes. Indicate kilobytes with `k` or `K` and megabytes with `m` or `M`.

For example:

---

```
-s 1M  
-s 1024k  
-s 2000000
```

If the size parameter to `addvol` is not specified on the command line, then the default size will be used. If the `datavol` entry is already in the server profile file then, the volume with the same name cannot be created manually because the server assumes that the volume with this name already exists.

**NOTE:-** If the database is associated with password authentication mechanism, then `addvol` utility will prompt the user to specify the DBA password before making any changes to the database.

## Usage Notes for Unix users

- A database on UNIX can use either files or raw devices for the database volumes. If you are using files, they may be local or remote.
- If the database volumes are remote UNIX files, they may be accessed through NFS.

**CAUTION:-** However, we strongly discourage you from accessing database files through NFS, because the NFS protocol does not guarantee that file writes are flushed to disk on invocation of a flush system call. This means that use of NFS could result in database corruption!

It is much better to create a database on the host where the file system is physically located and then access it with `VERSANT db@host` protocol rather than with NFS protocol. This allows a Versant server process to directly access the database files and will improve performance.

- If you want to use a UNIX raw device for a database volume, make sure that the partition you use does not include cylinder 0 or cylinder 1. When a UNIX partition is used for a raw device, all cylinders allocated to that partition would be used.
- If the UNIX partition is located on 'a' or 'c' ('c' implies the entire disk), the disk label can be overwritten. The workaround is to make partition 'a' two cylinders long and then start partition 'b' at cylinder 2. You can then safely use partition 'b' for raw devices.
- If you use a UNIX raw device for the physical and/or logical log volumes, you may run out of space, because raw devices cannot be dynamically expanded by Versant. By contrast, if you use files for the log volumes, Versant will increase the log volume sizes when necessary. Of course, the log files cannot expand if their disk partition is full.
- In order to create database volumes on UNIX raw devices, the person who creates the database, should be made the owner of the device, and be given the database owner read-write permissions.

---

For more information refer to “Database Creation Basics” on page 54, in “Chapter 2 - Database Creation”.

## cleanbe

The `cleanbe` system utility present in the Versant `bin` directory runs itself periodically to clean up unused resources at the backend.

**CAUTION:-** Do not invoke this utility explicitly. It is only documented here because of its visible presence in the Versant `bin` directory.

## cleanfe

The `cleanfe` system utility present in the Versant `bin` directory runs itself periodically to clean up unused resources on the frontend.

**CAUTION:-** Do not invoke this utility explicitly. It is only documented here because of its visible presence in the Versant `bin` directory.

## comparedb

### compardb

```
UNIX    comparedb [options] db1 db2
WIN     compardb  [options] db1 db2
```

This utility counts the objects in the databases specified as `db1` and `db2`, confirms that the logical object identifiers for the objects in the databases are the same, and displays the logical object identifiers for the first object that is different.

This utility is useful in comparing the contents of a primary and synchronous replication database.

Before using `comparedb`, you must use the `dbinfo` utility to set both databases to single user mode in order to compare stable database states.

**See also Dbinfo utility.**

The second comparison database `db2`, must be a group database.

You can call `comparedb` from a program by using the C/Versant `o_comparedb()` function or the C++/Versant `comparedb()` method.

### Options for Comparedb Utility are:

The default option is:

**-classes <class1><class2> ...**

Compare objects of only the specified classes.

**-fullcompare**

Display list of loids of objects, which are different in two databases. (This option is incompatible with `-noprint` option).

**-noprint**

Suppress display messages while running.

**-value**

Compare object values as well as object identifiers and display the first logical object identifier for objects with non-equal values.

**-advanced -classes**

Compare objects of only the specified classes with alternative algorithm. This option can be used with classes having large number of instances.

**NOTE:-** If any or both the databases are associated with DBA password then the user will be prompted for their respective passwords to complete the DBA authentication. If both databases need DBA authentication, the two databases should have the same DBA password. If only one of them needs DBA authentication, then the DBA should give the password for the database which needs DBA authentication.

## convertdb

## cnvrtdb

UNIX    `convertdb [options] database_name`

```
WIN      cnvrtdb [options] database_name
```

This utility converts one Versant database version with another. An existing database created with a prior Versant release is converted to the current Versant release and returns an error if not successful.

The system prompts a confirmation message before starting the conversion procedure. Enter “yes” to continue with the conversion of the database or “no” to abort the operation.

### Indexing related changes

Due to indexing related changes, specific Btree indexes created prior to this release have to be recreated.

These indexes include attributes involving multiple object links (e.g:-LinkVstr<type>--&, LinkVstrAny), the btree/hash indexes on attributes involving o\_float, o\_double, o\_date, o\_time and o\_timestamp.

Convertddb generates a text file “deletedindex.dat”, located in the database directory, that contains the information about the indexes to be recreated.

Convertddb deletes the listed indexes as a part of conversion procedure, and leaves the index creation to the user. The system prompts a confirmation message before starting to delete the indexes. If you want to delete the indexes and continue with the conversion, key in “yes” or “no” to abort the operation.

During the conversion procedure, the listed indexes are deleted and a shell script “recreateIndex.sh” (or batch file “recreateIndex.bat” on Windows) is generated in a database directory. This shell script can be executed later to recreate the deleted indexes.

**IMPORTANT:-** Please make sure that you have backed up the database.

### Options for Convertddb Utility are:

**-noprint**

Suppress some display messages while running.

You must end all transactions and then stop the database with stopddb before and after converting it. Do not stop the database with the -f option to stopddb.

### Example:

```
convertddb pdb
VERSANT Utility CONVERTDB Version 7.0.1.3
```

---

Copyright (c) 1989-2006 VERSANT Corporation

**WARNING:**

Please make sure that the database is backed-up before proceeding with conversion.

Would you like continue? Enter "yes" to continue: yes

The indexes need to be deleted during the database conversion. Deletion is required due to change in index structure layout.

The list of indexes that will be deleted is stored in `"/usr/local/versant/db/pdb/deletedindex.dat"`

Continuing will delete all the listed index.

Would you like continue? Enter "yes" to continue: yes

Script `"/usr/local/versant/db/pdb/recreateIndex.sh"` is created in the database directory for regenerating the deleted indexes.

Deleting index on Class: CEmployee

Attribute: CEmployee::salary Index: Hash ...

Deleting index on Class: CEmployee

Attribute: CEmployee::salary Index: Btree ...

Deleting index on Class: CEmployee

Attribute: CEmployee::ID Index: Unique Btree ...

Deleting index on Class: CEmployee

Attribute: CEmployee::department Index: Btree ...

If the database contains `bilink` or `bilink vstr` objects created with `cfront 2.1`, you will need to run the `rebilink` utility after running the `convertdb` utility.

**NOTE:-** `convertdb` does not need DBA passwd.

## Conversion of Versant Database Release 6.x to Release 7.0

The following table briefly describes the supported conversion:

FROM (prior release)	TO (current release)	PROCEDURE
6.x (32-bit platform)	7.x (32-bit platform)	Run <code>convertedb</code> from 7.x (32-bit) Release.
6.x (32-bit platform)	7.x (64-bit platform)	Run <code>convertedb</code> from 7.x (64-bit) Release.
6.x (64-bit platform)	7.x (32-bit platform)	Not supported.
6.x (64-bit platform)	7.x (64-bit platform)	Run <code>convertedb</code> from 7.x (64-bit) Release.

## createdb

```
createdb [options] dbname
```

This utility creates, formats, and initializes a new database with the name `dbname` and either makes the new database a part of an existing distributed database system or starts a new database system.

**NOTE:-** Before using `createdb`, you must use the `makedb` utility to create a database directory. This will also create the backend and frontend profile files.

**See `makedb` utility.**

The database name `dbname` must be unique for the distributed database system to which it will belong.

You must be the owner of the database (DBA) or the super user to run this utility.

If the user has specified a password at the time of `makedb` operation, then the user is prompted for a DBA password and only then it will authenticate you to create, format and initialize a database.

For a remote database, append the node name to the database name using the syntax `database@node`.



---

This utility will create the physical log, logical log and system volumes using either values found in the server process profile or else with default values.

If the server profile contains `datavol` entries for data volumes, then these volumes will also be created.

**For more information on data volumes, refer to “Quick Reference” on page 130 and “datavol” on page 99 in “Chapter 3 - Database Profiles”.**

The `createdb` utility will update the `osc-dbid` file to register the new database with the existing distributed database system. The file `osc-dbid` containing the path and name of all databases in the system must be visible from your machine when you run `createdb`.

**Options for `createdb` utility are:**

The default option is:

**-i**

Reserve space for the system volume.

Reserving space can prevent you from running out of space at runtime. For example, suppose you set the size of the system volume as 100 megabytes in your profile, but you really have only 60 megabytes physically available. In such case, if you use `-i`, you will immediately get an out-of-space error, but if you do not use `-i`, you will get an error at runtime as the system volume dynamically expands.

If you have defined multiple storage volumes and the physical space in one volume is less than the logical space, you will get an out-space-error at runtime even though there is space available in the next volume.

Pre-allocating space may improve performance, because space will not have to be dynamically allocated. However, pre-allocating space will slow down the process of creating the database, and because the space is used immediately, it will prevent it from being used for other purposes.

The `-i` option has no effect on UNIX raw devices.

Space for log volumes is always pre-allocated according to the parameters set for the `plogvol` and `llogvol` size specifications.

**-il**

Reserve and pre-format file system space for the logical log volume and the physical log volume.

**-noprint**

Suppress display messages while running.

If the `-i` option is specified, both the system volume and any data volumes created will be initialized.

Examples of using `createdb`:

```
createdb mydb
createdb -i mydb
```

Key components of the newly created database will be:

## System volume

A system volume for catalog information and data storage, with a name, location, size, and device according to the specification in the server process profile file `profile.be`.

The default is a file of 128 megabytes, named 'system' located under the database directory. For example, for a database named `dbname`:

```
/usr/local/versant/db/dbname/system
```

## Physical log volume

A physical log volume for physical data information related to logging and recovery, with a name, location, size, and device as specified in the server process profile file `profile.be`.

The default is a file of 2 megabytes named `physical.log` located under the database directory. For example, for a database named `dbname`:

```
/usr/local/versant/db/dbname/physical.log
```

## Logical log volume

A logical log volume for transaction undo-redo information related to logging and recovery, with a name, location, size, and device as defined in the server process profile file `profile.be`.

The default is a file of 2 megabytes named `logical.log` located under the database directory. For example, for a database named `dbname`:

```
/usr/local/versant/db/dbname/logical.log
```

## .sharemem

A shared memory file, which contains a shared memory identifier.

---

For more information refer to “Database Creation Basics” on page 54, in “Chapter 2 - Database Creation”.

## createreplica

### creatrep

```
UNIX    createreplica [options] primary_db replica_db
WIN     creatrep      [options] primary_db replica_db
```

This utility creates or designates a replica database and then loads the replica database with the contents of a primary database.

**NOTE:-** You must stop the databases before using this utility.

**See** `stopdb` utility.

For database sizes in Gigabytes, `profile.be` parameters of `replica_db` should be tuned properly so that `createreplica` takes reasonable time.

You must use this utility before beginning replication with the Fault Tolerant Server option.

Before running this utility, you must first create a text file named `replica` in your software root directory (the directory specified by the `VERSANT_ROOT` parameter) and add a line of the following form:

```
primary_db@node1 replica_db@node2
```

If, at runtime, this utility does not find the text file named `replica` and/or does not find an entry for the primary database, then this utility will fail.

Both replica databases must be group databases. If either or both the replica databases are associated with a password during utilization of `makedb`, then the utility requires their DBA passwords for authentication.

You can only create one replica for a database.

The `createreplica` utility will create storage and log volumes and update the network database identifier file `osc-dbid` to register the new database as part of the current database system.

For `createreplica` to update the `osc-dbid` file, the `osc-dbid` file must be accessible from your machine at the time you run `createreplica`. If you do not know the location of the `osc-dbid` file for the source database, you can use the `oscp` utility to find it. If you have multiple database systems

in your network, you may also want to confirm that your system configuration files or environment variables point to the correct `osc-dbid` file.

**See also “oscp” on page 203, and “Database System Identifier File (osc-dbid)” on page 55 in “Chapter 2 - Database Creation”.**

You can call `createreplica` from a program by using the C/Versant `o_createreplica()` function or the C++/Versant `createreplica()` method.

The synchronous replication database created with `createreplica` is a normal database like any other. This means that you can use normal database utilities with the synchronous replication database.

For example, if the synchronous replication database is not large enough to hold the data in the primary database, you will receive an error. To increase the size of the synchronous replication database, follow normal procedures to add additional storage space with the `addvol` utility.

**See also “Quick Reference” on page 130.**

**For more information on synchronous database replication, please refer to the *Versant Wedding Usage Guide*.**

## Parameters are:

### `options`

Options for `createreplica`.

### `primary_db`

The name of the database to be replicated. Use `db@node` syntax to specify a remote database.

The primary database must be a group database.

### `replica_db`

The name of the synchronous replication database. Use `replica_db@node` syntax to specify a remote database.

The replica database must be a group database.

## Options for `creatrep` Utility are:

`-i`

---

Create, pre-allocate and initialize space for the synchronous replication database.

If you specify `-i`, then, before running this utility, you must create the database directory and support files by running `makedb`.

If the replica database already exists, you will get the error `UT_DB_EXISTS`.

#### **-nocreate**

Do not create the target database.

If you specify `-nocreate`, then, before running this utility, you must have previously created the target database.

An error will be returned if:

- The target database does not exist.
- The user does not have write access to the target database.
- The target database does not have enough space allocated to it to contain all objects in the origin database.
- The target database already contains user class definitions or objects. In this case, the error raised will be `UT_DB_NOT_EMPTY`.

#### **-threads N**

Copy a database using `N` threads. The number of threads is limited by system resources. Increasing `plogvol` and `max_page_buffs` in `profile.be` and turning off locking will also improve performance. If this parameter is not specified, `createreplica` will optimize and create 3 to 30 threads by itself.

#### **-noprint**

Suppress display messages while the utility is running.

#### **-batchsize M**

The default is 1000.

Copy `M` objects per transaction. In case there are large objects, there is a chance that user gets out of heap errors. You can use `-threads` and `-batchsize` option to keep the heap requirement low. If not specified `createreplica` takes 1000 objects per transaction.

## **-noprogess**

Do not report the overall copy progress on this console. By default, the overall progress is displayed on the console via a progress guage.

## **db2tty**

```
db2tty -D dbname [ options ] [ classnames ... ]  
db2tty -D dbname [ -l ] -o loids
```

This utility displays the contents of the database specified as `dbname`.

In the first form, you can optionally specify the names of classes whose contents you want displayed. In the second form, you can specify the objects you want displayed by using their logical object identifiers (loids.)

**NOTE:-** Please make sure that the `ss.d` is running before making use of this utility.

### **Options for db2tty Utility are:**

The default option is:

#### **-a**

Show all classes, including system classes.

#### **-i**

Show instances.

#### **-l**

Obtain a read lock on the objects to be displayed. The default is no lock.

#### **-n**

The number of objects to be retrieved at a time. The default cursor batch size is 200.

This option can be only used with the `-i` option.

#### **-o loids**

The logical object identifiers of the objects to be displayed.

## **-p**

Password corresponding to the user name entered in `-u`. This is mainly intended for scripts so that users don't have to enter the password interactively. For normal usage, users can just pass in the `-u` option and `db2tty` will prompt them for the password. Passwords cannot be more than 256 characters in length.

Running with just the `-p` option without the `-u` option will result in an error.

## **-s**

Show schema information only.

## **-u**

User name to access the DB. This user has to be a valid user of the DB . You'll be prompted for a password if no password is entered using the `-p` option. Username cannot be more than 32 characters in length. The default is to run as the OS user without any password.

**NOTE:-** If the server profile parameter `commit_delete` is ON and `db2tty` is run at that time when some objects have been marked for deletion by other transactions (which are not yet committed), then the instance count of these classes could be incorrect. For performance reasons `db2tty -d <dbname>` will return the instance count of objects physically present in the database at that moment.

If you need to count the actual number instances present in the database run `db2tty -i -d <dbname>`.

## **dbid**

```
dbid [options] <dbname>
```

This utility manages the `osc-dbid` - database system identifier file.

The `osc-dbid` file stores the route path and name of all databases in a particular database system.

**For more information refer to “Database System Identifier File (osc-dbid)” on page 55, in “Chapter 2 - Database Creation”.**

The `dbid` utility is used by the `createdb` and `removedb` utilities.

**See also** `createdb` and `removedb` utilities.

**NOTE:-** Normally, you should not invoke `dbid` explicitly unless directed to do so by Versant Customer Support.

**See also “setdbid” on page 217.**

**Options for `dbid` utility are:**

**-N**

Create an `osc-dbid` file. `<dbname>` is not specified.

**-c**

Create an entry in `osc-dbid` file for database `<dbname>`

**-C `<dbid>`**

Create an entry in `osc-dbid` file for database `<dbname>` and database id as `<dbid>`.

**-t `<type>`**

Specify the type of database (group or personal). This option must be used with `-c` or `-C` options. Default database type is group.

**-d**

Remove the database `<dbname>` entry from the `osc-dbid` file.

**-r**

Print the database id of the database `<dbname>`

**-noprint**

Suppress display messages.

**CAUTION:-** The “-c” and “-C” are advanced options and should be used with great care. If they are to be used then please ensure that you give a subsequent call to `setdbid` with the `dbid`, which has just been created.

## dbinfo

`dbinfo option database_name`



---

This utility determines or sets the current mode of the database specified as `database_name`.

This utility can be used only by the database administrator (DBA) of a database. Along with the OS check, the password-based authentication of the DBA will also be done if the database is associated with a password.

**Parameter for dbinfo utility are:**

**-c**

Create a new `.lock` hidden file in the current directory.

**-p**

List the current database mode.

[ -m | -0 | -1 | -d | -r ]

**-m**

Set database to multi-user mode.

This is the "normal" mode for a database.

For a group database, this means that, per the "normal" definition of a group database, any number of users can make any number of connections.

For a personal database, this means that, per the "normal" definition of a personal database, only one user at a time can use it and can use it only as a session database.

**-0**

Set database to `unstartable` mode.

This mode prohibits all access to the database and prevents it from being started with the `startdb` utility tool.

**-1**

Set database to `dba/single-connection` mode.

The database is startable, but it will only accept a single connection from a single client run by the database administrator.

**-d**

Set database to `dba/multiple connections` mode.

**-r**

Set database to read-only mode.

The database is startable. However, all data volumes and log volumes are opened in read-only mode and cannot be modified. If you want to access the database from a CD/DVD, then you need to change the mode to read-only before writing the database files to the media. Once the database is written on the CD/DVD you can mount it on the desired machine and you can change the Versant database directory in your environment to point to the media directory that contains the read-only database.

**For more information on how to change your Versant database location please refer to chapter 6 “Configuration Parameters” on page 315.**

Once this is setup, you can start the database and use it for all types of read-only transactions.

## Restrictions and Suggestions for Read-only mode

- If a transaction tries to modify any data for a read-only database then it will get the Versant error 2971 - SM\_DB\_IS\_READONLY.
- You cannot change the database mode to read-only mode if database is a part of FTS pair. If you try to do this, you will get the Versant error 2973 -SM\_DB\_IS\_REPLICATED.
- The database needs to be stopped before changing the mode to read-only or when changing the mode from read-only to some other database mode. If the database is not stopped and you try to change the mode as explained above then you will get the Versant error 2972 - SM\_DB\_IS\_ACTIVE.
- It is strongly recommended that the database be stopped using the normal `stopdb <dbname>`. When a read-only database is started up, it does not do physical and logical recovery. If the database was stopped forcefully with a `stopdb -f`, for example, then there would be no checkpoint and the log volumes will contain some data. Hence, it is possible that you might see inconsistent data after database startup as the database might not be transactionally consistent. Therefore, it is recommended to do an orderly shutdown so that the log volumes are cleared via a checkpoint.
- A read-only database on CD/DVD can only be started by the DBA of the database. The Database Administrator (DBA) of the database that is on CD/DVD is determined the same way as it is done for databases that are on normal disk drives.
  - On UNIX, the owner of the database directory is the DBA therefore, for a read-only database on CD/DVD the DBA would be determined by the mount options used to mount the CD/DVD device on the machine. For example, if the device is mounted in such a

way that “root” owns all the files then only “root” can startup the database as he will be the DBA. This also means that the DBA of a read-only database can be changed by changing the way the device is mounted.

- On Windows, the DBA is decided by the content of the `Personal.flg/Group.flg` file in the database directory. So once a read-only database has been written to the CD/DVD the DBA cannot be changed as the DBA can only be the user whose name is present in `.flg` file. Therefore before writing a database to CD/DVD care needs to be taken that the username in the `.flg` file is the name of the user who will access this database through the CD/DVD. For example, if user “Larry” creates and writes a database onto a CD and he intends that it will be used by user “Vishal” then before writing the database to the CD he needs to modify the `.flg` file in the database directory and change the entry from “Larry” to “Vishal”. Once this is done, only “Vishal” can start this database from the CD and not “Larry” or any other user.
- It is possible to place more than one database on the CD/DVD in various folder hierarchies. If there are multiple databases on the CD/DVD in different directories then in order to access them, the Versant database directory in your environment needs to be changed accordingly. For example on Windows, if your database name on the CD is “gdb”, your CD drive letter is “D:” and the database path is “D:\gdb” then your Versant database directory should be set to “D:\” in order to access “gdb”. If there is another database “pdb” whose path is “D:\subfolder\pdb” then in order to access “pdb” your Versant database path should be set to “D:\subfolder”.
- Access to the database system identifier file, `osc-dbid`, is not supported when the file is on a CD/DVD.

The `dbinfo` utility may be invoked either at the command line or through the `o_dbinfoapi()` function. If conflicting options (`-m`, `-0`, `-1`, `-d` and `-r`) are given, an error will be thrown.

You can change a database mode from one state to another at any time (except when changing the mode from/to read-only). After a new mode is set, it only affects new connections, and existing connections run without interruption. If you want to break existing connections, after running `dbinfo`, you must stop the database with `stopdb`.

One purpose of the `dbinfo` utility is to allow a way to guarantee that the DBA is the only user of a database during a database reorganize operation with the `reorgdb` utility. To guarantee that the DBA is the sole user, run `dbinfo` to set the database mode to either `dba-only` or `dba/single-connection`.

## dblist

```
dblist [options] [@node]
```

This utility lists the databases in a system of databases coordinated by an `osc-dbid` database system identifier file.

To run `dblist` on a remote node, specify the name of the remote machine using the optional `@node` argument, where `node` is the name of the remote machine.

If there are multiple database systems in your network, the databases listed will be those in the network database system coordinated by the `osc-dbid` file specified in the currently set environment parameters `VERSANT_DBID_NODE` and `VERSANT_DBID`.

### Options for dblist Utility are:

**-all**

This is the default option.

List all databases in the database network system coordinated by the `osc-dbid` database system identifier file specified in the currently set environment parameters `VERSANT_DBID_NODE` and `VERSANT_DBID`. This is the default option.

**For more information on the search procedures that are used to determine the current database system, See also “Installed Directories and Files” on page 282 in “Chapter 5 - Directories and Files”.**

**-owner owner\_name**

List all databases in the system owned by `owner_name`.

**-d dbname[@node]**

List only the named local or remote database.

For a remote database, append the node name to the database name using the syntax `database@node`.

**-dir**

List all database directories, including directories that do not currently contain a database, branching from the database root directory specified by the current setting of the `VERSANT_DB` environment parameter plus all databases that would be returned by the `-all` option.

---

**-noprnt**

Suppress display messages while running.

**Examples:**

```
dblist
dblist -all
dblist -owner myname
dblist -dir
```

**Sample output:**

```
VERSANT Utility DBLIST Version 7.0.1.3
Copyright (c) 1989-2006 VERSANT Corporation
ID                = 4678
DB name           = db01@server1
creator           = cp
date created      = Thu Jul 14 22:19:40 2005
db type           = GROUP DATABASE
db version        = 7.0.1.3

ID                = 4679
DB name           = db02@server2
creator           = cp
date created      = Thu Jul 14 22:19:49 2005
db type           = GROUP DATABASE
db version        = 7.0.1.3
....
```

## dbtool

```
dbtool <module> [options] <dbname>
```

Depending upon options specified, for a specified database you can use the `dbtool` utility to:

- Get information about locks
- Get information about an object
- Get information about transactions
- Get information about database system

- Get information about storage volumes
- Get cluster instances of a class
- Get information about indexes
- Create and delete index
- Check consistency and maintain AT table
- Trace components of a database
- Get event notification information
- Get information about all the classes
- Get information about logging activities
- Determine fragmentation information and free space in database volumes
- Get information about schema evolution
- Explicitly evolve all instances to the latest Schema

This utility can be used only by the DBA (owner) of the specified database. It cannot be called using an interface routine.

**NOTE:-** Along with the OS check, the password-based authentication of the DBA will also be done if the database is associated with a password during creation.

Described below are the options for `dbtool`, arranged by functionality:

## Get information about a Lock

```
dbtool -locks <options> <dbname>
```

### Options for this utility are:

#### **-info**

Prints information about current locks held on objects in the specified database. See the `-table` option for more information.

#### **-table [-c connid ...] [-o loid ... ]**

Prints lock information for a set of given connections and objects for the specified database.

For `connectionid..`, you can specify one or more connection identifiers.

---

For `loid..`, you can specify one or more logical object identifiers in dotted triple notation.

You can use either or both of the `-c` and `-o` options.

If neither `-c` nor `-o` are supplied, `-table` is similar to `-info`, except that `-table` supplies more information than `-info`.

For example, for a database named `group`:

```
dbtool -locks -table group
dbtool -locks -table -c 1 group
dbtool -locks -table -o 1.0.23456 group
dbtool -locks -table -c 1 -o 1.0.2345 group
dbtool -locks -table -c 1 2 -o 1.0.12345 1.0.45678 group
```

**For each connection, the following will be printed:**

- the connection identifier
- information about the connection, including client hostname, client process identifier, and user-name.

**For each locked object, the following will be printed:**

- the logical object identifier (loid) and class name
- the lock mode
- Lock flags indicating whether the lock has been granted ("Hold") or being waited for ("Wait").

## Sample Output

Following is sample output that shows information for all locks held on objects in a database named `db2`. Note that there are two users.

User `venka` is on a machine named `alpha` with a Versant connection id of 6 and a system process id of 28737. User `venka` holds 6 short locks and is waiting for 1 short lock. User `venka` is in a session named `VersantView-28737`, which is a default name created by the Versant View browser program based upon the process id.

User `George` is on a machine named `beta` with connection 10 and process 28742. User `George` holds 7 short locks. User `George` is in a session named `VersantView-28742`.

In this case, most of the locks are held on objects of class `Course`. Locks indicated as an instance of class are on a class object. Locks indicated as an instance of attribute are on an attribute object.

**NOTE:-** The type of short lock is indicated by the usual symbols for lock mode parameters (WLOCK = write lock, RIWLOCK = read/intention-write lock, IWLOCK = intention-write lock, ULOCK = update lock, RLOCK = read lock, IRLOCK = intention-read lock, and NOLOCK = null lock.)

```
% dbtool -table db2
```

```
Connection Id: 6
```

```
Client: alpha[pid: 28737] Session: VersantView-28737
```

```
User: venka
```

LOID	[instance of]	Lock	Lock Flags
347.0.162850	[Course ]	RLOCK	Hold,Transient
347.0.162849	[Course ]	RLOCK	Hold,Transient
347.0.162848	[Course ]	WLOCK	Wait,Transient
347.0.162822	[attribute ]	RLOCK	Hold,Transient
0.0.2	[class ]	IRLOCK	Hold,Transient
347.0.162821	[attribute ]	RLOCK	Hold,Transient
347.0.162820	[class ]	IRLOCK	Hold,Transient

```
Number of locks: 7
```

```
Connection Id: 10
```

```
Client: beta[pid: 28742] Session: VersantView-28742
```

```
User: George
```

LOID	[instance of]	Lock	Lock Flags
347.0.162850	[Course ]	RLOCK	Hold,Transient
347.0.162849	[Course ]	RLOCK	Hold,Transient
347.0.162848	[Course ]	RLOCK	Hold,Transient
347.0.162822	[attribute ]	RLOCK	Hold,Transient
0.0.2	[class ]	IRLOCK	Hold,Transient
347.0.162821	[attribute ]	RLOCK	Hold,Transient
347.0.162820	[class ]	IRLOCK	Hold,Transient

```
Number of locks: 7
```

Following is sample output for an object with loid 347.0.162849 in database db1. In this case, user venka has a short read lock on this object and user George has an update lock and is waiting for a write lock on the same object. From the point of view of a database, each transaction has only one lock entry per object, so in this case, even though George has an update lock, what you will see is an entry showing the wait for a write lock.

```
% dbtool -table -o 347.0.162849 db1
```

```
Connection Id: 7
```

```
Client: alpha[pid: 28790] Session: VersantView-28790
```



```
User: venka
LOID      [instance of]  Lock      Lock Flags
=====
347.0.162849 [Course      ]  RLOCK      Hold,Transient
Number of locks: 1
Connection Id: 22
Client: beta[pid: 28814] Session: VersantView-28814
User: George
LOID      [instance of]  Lock      Lock Flags
=====
347.0.162849 [Course      ]  WLOCK      Wait,Transient
Number of locks: 1
```

## **-stats**

Prints lock statistics.

Following is sample output for a database named group that shows lock statistics since the moment when the database was started.

```
dbtool -stats group
** Lock Statistics **
Locking statistics:
    total = 98443
    outstanding = 97659
    deadlocks = 0
    conflicts = 0
    requests = 196101
    objects = 97659
```

Following is an explanation of the lock statistics fields:

**Total:** The number of locks granted since the database was started.

**Outstanding:** The current number of locks granted (a particular object may have more than one lock; for example, an object might have numerous read locks.)

**Deadlocks:** The number of deadlocks that have occurred since the database was started.

**Conflicts:** The number of times a lock request has been blocked since the database was started.

**Requests:** The number of lock requests since the database was started.

**Objects:** The number of locked objects.

## Get information about an Object

```
dbtool -object [operation] [options] <dbname>
```

**-info <x.y.z>**

Prints information about the object with the specified logical object identifier (LOID) <x.y.z>.

The information printed is useful primarily to Versant Technical Support, as it indicates information about the internal workings of Versant.

### Sample Output

```
dbtool -object -info 2142.0.7187 group
LOID      : [2142:0:7187] is in AT and in data page
           {Rvolid=0, Rpage=70, Rslot=114}
KIND      : Normal object
CLASS     : class
SIZE      : 248
STATUS    : Normal
TYPE      : Normal, never moved.
```

## Get information about Transactions

```
dbtool -trans [operation] <dbname>
```

**-info**

Prints information about current transactions involving objects in the specified database.

**For example:**

```
dbtool -trans -info group
** Transaction Info **
```

---

transaction ID	Coord	count	flags	pid.tid	name	long trans ID
2.0.428046	y	00000	x0290	22976.5616	checkdb	
2.0.428048	y	00000	x0290	22978.7424	dbtool	

Following is an explanation of the transaction information provided:

**Transaction ID:** Transaction identifiers assigned by Versant. This information is for use by Versant Technical support.

**Long trans ID:** Has been deprecated.

**Coord:** an indication of whether this database is storing coordinator information for two-phase commits (in other words, whether the specified database is being used as a session database.)

**Count:** the number of locks held by current transactions on this database.

**Flags and pid.tid:** these columns provide internal information useful to Versant Technical Support.

**Name:** the applications with a current connection to the database.

## Get information about Database System

```
dbtool -sys [options] <dbname>
```

**Options for this Utility are:**

**-check [-MT] [-v] [-inst]**

Perform an overall database consistency check. This option can be used independently or in combination with `-MT` or `-v` or `-inst`.

**-MT**

Enable consistency check in multi-thread mode. This option is useful only for consistency check, not all options of `dbtool` can be run in multi-thread mode.

In order to use this option database has to be in DBA multi-connection mode.

If this option is used in conjunction with '-v (verbose)' mode, `dbtool` verbose output is created in following files:

`checkAT` for AT table check output

`checkdb_x` (where 'x' could be from '0...n', where 'n' represents number of threads working on class segments).

These files will be created in the location where `dbtool` is invoked OR if an environment variable '`VERSANT_DBTOOL_DIR`' is present, then these files will be created in the directory specified by environment variable.

Whenever '`dbtool`' reports inconsistencies, these files need to be examined for more details.

**-v**

Verify database consistency in verbose mode.

**-inst**

Perform overall database consistency check and object consistency check. If an object is found in corrupted state, it prints the corresponding loid, poid, number of corrupted loids in a class and finally reports inconsistencies detected during the scan.

**-info -activedb**

Get names of all currently running databases located under the setting for database root parameter `VERSANT_DB` in use by the current process. If there are multiple database root directories on the machine, the names of databases running under other root directories will not be returned.

The information returned is the same as returned by the C/Versant function `o_getactivedblist()`.

**-info -resource**

Print system usage of shared memory, processes, threads, and connections that are associated with the specified database.

## Sample Output

Following is sample output.

In the case of "Processes and Threads" table, the output for "user@client.." column has been wrapped to the next line.

```
dbtool -sys -info -resource group
Resources being used by the database: group
Shared Memory:
The following segments are in use:
  1  43000
  2  43001
Processes and Threads:
PID      TID      ConnID  State  Type
User@client[IP address:port], clientpid[session]
-----
28808    1         19     alive  Vbackup
28788    1         2      alive  Cleanbe-proc
28789    1         3      alive  Single-proc
venka@alpha[single process],28789[test1]
28792    1         5      alive  Obe-proc
28792    4         6      alive  Cleanbe-thrd
28792    5         7      alive  Obe-thrd
venka@alpha[192.70.173.172:37109],28790[test2]
```

## Shared memory

The "Shared Memory" section shows the shared memory segments in use by database group.

## Processes and threads

The "Processes and Threads" section provides information about the applications and system processes in use by database group. The columns are:

**PID:** The operating system process identifier.

**TID:** The operating system thread identifier.

**ConnId:** The connection identifier assigned by Versant.

**State:** Whether the thread or process is "alive" or "dead".

**Type:** The type of thread or process. The following codes are used:

- Obe-thrd, a database thread that serves an application
- Single-proc, an application running as a single process
- Cleanbe-proc, a database cleanup process
- Obe-proc, a database management process

- Cleanup-thrd, a clean-up thread running within the server process
- Vbackup, a process used by the vbackup utility
- Special, a special process, such as an event daemon

## Client information

There is a "Client information" section only if the thread or process is associated with database connection made by an application (in the above, the client information is shown as wrapped to the next line.)

The client information will include the following:

**Username@hostname:** The user and machine names associated with the application that has a database connection.

**IP address:port number:** If the application is running in a single process, you will see [single process]. If the application is running in a different process, you will see the system TCP/IP address number (such as 192.70.173.172 in the above example) and port number (such as 37109) for the application.

**Process ID:** The operating system process identifier (such as 28790 in the above example.)

**Session name:** The name of the Versant session (either a specified or default name.)

## Get information about Storage Volumes

```
dbtool -volume [operation] [options] <dbname>
```

**-info -mount**

List the names, sizes, and locations of all data storage volumes associated with the specified database.

**For example:**

```
dbtool -volume -info -mount group
Volume 0:
Sysname "sysvol"      Size: 131072K
Pathname "/net/vp/mvp/lang/george/versant/db/group/system"
```

In the above case, the database group has one data storage volume named sysvol. It is 131,072 K bytes in size and is located at:

```
/net/vp/mvp/lang/george/versant/db/group/system.
```

## Cluster instances of a Class

```
dbtool -cluster <options> <dbname>
```

### Options for this Utility are:

```
-class <class1> <class2> [<class3> ...]
```

Clusters instances of the given set of classes in a particular database.

This action does not force a clustering. Instead, it suggests to the database that a clustering should be performed. If a class already has been assigned storage, no action will be taken for that class.

(description different in installer utility )

If any of your classes already have instances, the system returns this error:

```
6074, SCH_CLS_MUST_BE_EMPTY: Class must be empty to be clustered
```

**For more information refer to `o_cluster()` function in the C/Versant Reference Manual.**

## Get information about Indexes: create and delete index

```
dbtool -index [operation] [options] <dbname>
```

### Operations for this Utility are:

```
-check [<C> [<A>] ]
```

Verifies index consistency from class C towards index A where A is an index attribute under class C. C++ users specify C::A for an attribute.

```
-info {options}
```

Prints index information for classes and attributes in a database.

## Options for this Utility are:

### **-list [sub-options]**

Lists information about the index, optionally for just a class and/or attribute of a class.

All indexes for a database are displayed if no options are specified. No index information is displayed if no index exists. Names are case sensitive.

If the class was created with C++, you must use the precise attribute name.

### For example:

For the Name attribute of class `Employee` in database group:

```
dbtool -list Employee Employee::name group
Following is sample output for a database named group.
dbtool -list group
Class: FormNode           Attribute: ten
Domain: o_2b              Length: 2
Index: BTREE
Creation time: Thu Jul 14 21:45:49 2005
Number of pages: 2
Located in volumes:
/net/vp/mvp/lang/george/versant/db/group/system/sysvol
```

## Sub-Option Parameter for this Utility are:

### **-short**

Print index information for all classes and attributes in a database by component. This is the same as using `-list` without options, except that the display format is different.

### For example:

```
Employee    ssn    o_u1b    11    Btree
```

Information that does not fit in the tabular display does not appear.

<C>

All indexes for a class are displayed only if a class name is specified.



---

<A>

Indexes for an attribute are displayed only if a class and attribute are specified. Thus, the usage would be:

```
dbtool -index -info -list <C> <A> <dbname>
```

## Create Index

**-create {options} [options] <C> <A>**

Creates index for the specified attribute of the specified class.

### Options for this Utility are:

{-btree | -hash} [-unique]

#### **-btree**

This option is used to create a B-tree index for a specified attribute on a specified class.

#### **-hash**

This option is used create a Hash index for a specified attribute on a specified class.

#### **-unique**

This option is used create a B-tree or a Hash index and enforce unique values for the specified attribute.

## Delete Index

**-delete {options} <C> <A>**

Deletes an index for the specified attribute of the specified class.

### Options for this Utility are

#### **-btree**

This option is used to delete a B-tree index for a specified attribute on a specified class.

#### **-hash**

This option is used delete a Hash index for a specified attribute on a specified class.

## Load Index

**-index -load <C> <A>**

To load a B-Tree index into the server buffer, where C is the Class name and A is the attribute name. C++ users specify `class::attribute` for attribute name.

## Viewing consistency and maintenance of AT Table

`dbtool -AT [operation] [options] <dbname>`

AT table related options.

**Operation Parameter for this Utility are:**

**-AT -analyze**

Analyzes the AT (loid2poid table) for fragmentation.

**-AT -check [options]**

Verifies AT table consistency.

**Options for -check parameter are:**

**-v**

Verifies consistency of the AT table in verbose mode.

**-alloc x**

Pre-allocates space in the server process AT table for the number of entries specified as X.

This option can be used for performance tuning a large database.

**-AT -info -loids [options]**

Prints all the loids used in the system.

---

**-AT -load**

AT table maintenance-related options.

**-load**

Pre-loads the AT into the server buffer cache.

**<dbname>**

Database name.

**Options for -Version parameter are:****-AT -version -print**

This will print the current version of AT table.

**-AT -version -update**

This will update AT table to latest version.

This option has to be run on offline database, else will give an error message.

If database is not offline, an error message -"Database must be in DBA - only single user mode" appears.

If database is already up-to-date, it will give a message "No update required, AT version is already up-to-date; current version is 4"

In 701 service release 3, enhancements are made to AT table to use discontinuous memory.

Any new database will be created with new enhanced AT table, but old databases will continue to use old AT algorithm.

Old databases can be shifted to use new AT version using this option.

## Trace components of a database

You can use tracing to monitor the database server activity.

The primary purpose of tracing is to help Versant Technical Support debug your application.

Turning tracing `ON` will affect your performance.

Trace entries are written to a tracing log file. You can also display tracing output on `stdout`. You can allow concurrent threads to share the same trace file to preserve timing as much as possible.

You can turn tracing **ON** or **OFF** and specify tracing options either with server process profile parameters or with options to **dbtool**.

If you use **dbtool**, you control tracing while the database is running. Parameters used with **dbtool** will override parameters set in the server profile.

For example, if the profile sets tracing **ON**, you can use **dbtool** to turn it **OFF**. Options set with **dbtool** persist only while the database is running: when the database restarts, tracing parameters are reset to their values in the server profile.

The options that control tracing are:

```
dbtool -trace [<options>] [<command>]
```

### Options for this Utility are:

**-file <path>**

To view trace records from the file specified by path.

**-database <dbname>**

To view trace records from the trace file associated with the specified database.

If **-trace** is invoked with no options parameter, **dbtool** will look for the trace file specified by the **VERSANT\_TRACE\_FILE** environment variable.

### Command Parameter for this Utility are:

**-status**

Lists components plus **ON/OFF** status.

**-on [<components>]**

Turn **ON** specified components. If no components are specified, all components are turned **ON**.

**-off [<components>]**

Turn **OFF** specified components. If no components are specified, all components are turned **OFF**.

**-create [<entries>]**

Creates a new trace file with the specified number of entries in it. If number of entries is not specified, the default of 10,000 will be used.

**-view [<components>]**

Displays the contents of the trace file, showing entries for the specified components. If no components are specified, then trace records for all components are shown.

**-tail [-f] [-count <n>] [<components>]**

Displays last <n> entries in the trace file for specified components.

If the -f option is specified, additional entries will be shown as they are created.

If the -count option is omitted, then the last 20 trace records for all components are shown.

Invoking -trace with no command parameter is the same as using the -view command.

## Example:

The following is an example of the status command. Note how all components are on except "net":

```
% dbtool -trace -database example -status
```

COMP	DESCRIPTION	ON	COMP	DESCRIPTION	ON
====	=====	==	====	=====	==
am	Access Method	ON	at	AT Table	ON
bf	Buffer Manager	ON	bt	B-tree Manager	ON
dr	Dir Service	ON	hs	Hash Indexing	ON
io	I/O Sub System	ON	lg	Log Services	ON
ps	Process Manager	ON	rc	Record Manager	ON
sd	Storage Man SD	ON	si	Storage Man SI	ON
sm	Storage Manager	ON	tr	TransactServices	ON
net	Network Layer	--	sl	System Library	ON
vsl	Virtual System	ON	ut	Utility Services	ON
om	Object Manager	ON	qry	Query Manager	ON
lk	Lock Manager	ON	rcv	Recovery	ON

---

csr	Cursor	ON	ev	Event	ON
api	Kernel API	ON	rpc	RPC Messages	ON

**The following is another example of trace output.**

Each row represents a trace record. The "thread" field shows the process and thread which wrote the record (no entry indicates no change from the previous line). The "function" field shows the name of the internal function, which wrote the record. The "type" field indicates the type of the trace record (such as, an entry into the function or exit from the function). The "data" field shows additional record specific information.

```
% dbtool -trace -file /tmp/example.vtr -view
```

THREAD	FUNCTION	TYPE DATA
=====	=====	=====
26206.0001	net_tcpRead	rcvd dbid = 802, len = 16
	rpc_tr_commit	sent dbid = 802
	net_tcpWrite	sent dbid = 802, len = 256
	net_tcpRead	rcvd dbid = 802, len = 256
	rpc_tr_commit	rcvd dbid = 802
	rpc_tr_beglog	sent dbid = 802
	net_tcpWrite	sent dbid = 802, len = 256
	net_tcpRead	rcvd dbid = 802, len = 256
	rpc_tr_beglog	rcvd dbid = 802
	om_capi_term	exit api = o_xact
	om_capi_init	entr api = o_endsession
	rpc_tr_commit	sent dbid = 802
	net_tcpWrite	sent dbid = 802, len = 256
	net_tcpRead	rcvd dbid = 802, len = 256
	rpc_tr_commit	rcvd dbid = 802
	rpc_discon	sent dbid = 802

---

```
net_tcpWrite      sent dbid = 802, len = 256
net_tcpRead       rcvd dbid = 802, len = 256
rpc_discon        rcvd dbid = 802
om_capi_term      exit api = o_endsession
```

## Get Event Notification information

```
dbtool -event [options] <dbname>
```

### **-event**

Displays event notification for registration information, objects and processes.

You will see the following information:

- Event notification status - "active" or "not active".
- Event notification configuration information.
- Event message queue information, such as the number of events left in the event queue.
- A list of all event registration entries and their status - "active" or "not active".

## Get information about all Classes in a database

```
dbtool -class <operation> [options] <dbname>
```

### **Operation Parameters for this Utility are:**

```
-check { <class> | all } [options]
```

Checks a class for consistency.

**<class>**

Name of the class.

**all**

Checks all the classes.

### **Options for check is:**

**-v**

Verbose

**-inst**

Performs overall database consistency check and object consistency check. If an object is found in corrupted state, it prints the corresponding loid, poid, number of corrupted loids in a class and finally reports inconsistencies detected during the scan.

**-info [options]**

Prints information related to classes.

**Options for info are:**

**-list**

Prints all the class names, type and cardinality.

**<class>**

Name of the class.

## Get information about Logging Activities

All DBA activities should be logged to the per-database configurable file `LOGFILE` so that the history of commands can be viewed when required.

These activities may be those that make a physical change to a database or change the running state of a database.

```
dbtool -dbalog [<options>] dbname
```

**Options for this Utility are:**

The default option is :

**-setlevel <level>**

Sets the DBA logging level. The level can be either 0 or 1.

**-getlevel**

Prints the DBA logging level.



---

**-setfile <filename>**

Specifies the name of the file that will be used for logging.

Verify that appropriate OS permissions are given to create, modify or access the file. This file will be accessed from the database server machine.

**-getfile**

Prints the name of the file currently used for logging.

**The following utilities will be logged and will have two entries - one marked `START` and the other marked `END`.**

```
addvol
checkdb
convertdb
createdb
createreplica
dbtool
dropclass
dropinst
ftstool
habackup
removereplica
reorgdb
startdb
stopdb
vbackup
```

**The following utilities will be logged and will have only one entry marked `INVOKED`:**

```
dbinfo
dbuser
polling
setdbid
vconn
vstats
```

All the utilities that will be logged will have the `CWD`, `PID`, `RUID`, `EUID` and `HOST` fields.

**For more information refer to “LOGFILE” on page 297, in "Chapter 5 - Directories and Files".**

**The following is the list of tools/utilities that will not be logged into the per-database configurable LOGFILE file:**

```
cleanfe
comparedb
db2tty
dbid
dblist
expiry
itest
ivstats
makedb
makeprofile
oscp
removedb
udltool
vcopydb
verrindx
vstream
vtcl3g
```

## Determine fragmentation information and free space in database volumes

```
dbtool -space [options] <dbname>
```

Prints the free space in database volumes.

Also determines the degree of fragmentation of data segments by inspecting every page occupied in the database and appropriately advise the DBA for reorganization of data.

### Options for this Utility are:

```
-volume {sub-options} [sub-options]
```

Prints free space in volumes.

### Sub-option Parameters for this Utility are:

```
{-all | -n <name> -n <name>..}
```

---

Prints the free space available in the entire database counting all the volumes or prints the space details for each individual volume specified by `<name>` in the database. The output is printed both as a percentage of database (or volume) space free and number of kilobytes free.

**[ -n ]**

Specifies the volume name.

**[ -verbose ]**

Prints the details of free space available in each volume by using the `-verbose` option.

**[ -extraverbose ]**

Gives details of the free page-id's along with volume details.

**[ -pagespace ]**

Prints page free space in addition to extent free space. This option can be used on its own as well as with the `-verbose` and `-extraverbose` options.

**-segment {sub-options} [sub-options]**

Prints free space in the segments.

**-segment {-all | {-n <name> [-n <name>..]}} [sub-options]**

Determines the fragmentation information for segments of database `<dbname>`.

## Sub-option Parameters for this Utility are:

**-all**

If this option is specified then fragmentation information for all the segments will be inspected.

**-n <name>**

To specify a particular segment use this option.

More than one segment name can be provided by adding more `-n <name>` pairs to the command line. If the segment name does not exist, the error `SM_E_SEG_NOT_FOUND` will be thrown.

**-class {-all | {-n <name> [-n <name>..]}} [sub-options]**

Determines the fragmentation information only for the segment in which objects of any or a specified class reside. If the class name provided does not exist then the error `SM_E_CLS_NOT_FOUND` will be thrown.

More than one class name can be provided as follows:

```
dbtool -space -class -n <name> -n <name> <dbname>
```

**-fillfactor <percentage>**

Specifies the percentage of bytes occupied in a page that will render a page to be considered full. The default percentage is 70%.

**-sample [-size < sample size > ]**

The -sample option looks at 1000 random pages (can be changed with sub-option -size) taken as representative for all pages used by that class.

**-verbose**

Displays the allocation status of the pages of a segment. Pages that are empty (i.e., less than 10% utilized) will be marked 'E', pages that are full (i.e., more utilized than percentage specified in fillfactor) will be marked 'F' and other pages will be marked 'P' to denote partially full.

**-extraverbose**

Displays a graphical representation of pages of segment along with fragmentation information and PID of the pages. It prints the PID together with the fill grade character. A page id (PID) has the form, volumeid:pageid, for example, 0:1024.

**-highlevel**

Specifies that information should be gathered only at the page level without examining the individual pages.

**Example**

For example, to check the free space available in the database `spdb`.

```
dbtool -space -volume -all spdb
```

```
Percentage of free space in DB : 98%  
Total available free space in DB : 141120KB
```

For a more verbose output since this database has 3 additional volumes.

```
dbtool -space -volume -n <name> spdb
```

---

```

Volume 0:
Sysname "sysvol"      Size: 131072K
Pathname "/db/ara/spdb/system"
Percentage of volume space free in sysvol : 98%
Free space in vol sysvol : 129632KB
Volume 256:
Sysname "vol1"        Size: 4096K
Pathname "/db/ara/spdb/vol1"
Percentage of volume space free in vol1 : 93%
Free space in vol vol1 : 3840KB.
Volume 257:
Sysname "vol2"        Size: 4096K
Pathname "/db/ara/spdb/vol2"
Percentage of volume space free in vol2 : 93%
Free space in vol vol2 : 3840KB
Volume 258:
Sysname "vol3"        Size: 4096K
Pathname "/db/ara/spdb/vol3"
Percentage of volume space free in vol3 : 92%
Free space in vol vol3 : 3808KB
Percentage of free space in DB : 98%
Total available free space in DB : 141120KB

```

## Get information about Schema Evolution

```
dbtool -schevol <command> [-force] <C> <dbname>
```

Displays the history of schema evolution for a class <C>. Prints the schema of each class version indicating the attributes which have been added/deleted/modified.

### Options for command are:

#### **-mark**

Updates old class versions to indicate that they are fully evolved.

#### **-unmark**

Undo the changes done to the old class versions with the "-mark" command.

#### **-check**

Checks whether old class versions have any instances.

**-info [-last <n> | -all ]**

Prints the latest class version. Default will print the latest class version.

**-last <n>**

Prints the last n class versions.

**-all**

Prints all class versions.

**-force**

When used with "-mark" will update old class versions to indicate that they are fully evolved without checking for existing instances corresponding to them.

**<C>**

Class name

## Explicitly evolve all instances to the latest schema

VOD works with a lazy migration strategy i.e., it evolves instances of old schema versions to the latest schema version only when these instances are modified by database applications. This means that all instances get evolved over a certain time period depending on the access pattern. One disadvantage of having different instances for different versions in a schema is that it can impact query performance. In such a case, it is recommended to evolve all the instances to the latest schema version to speed up the queries on this schema.

This can be achieved using the server-side explicit instance conversion feature of VOD. This conversion can be triggered by the database administration tool: `dbtool`.

`dbtool -schevol` lists the schema evolution related options. The server-side explicit instance conversion feature is available under the "-evol" option.

### Options for command are:

**-evol**

Evolve all instances of old class versions to the latest class version.

---

**`[-fetchsize <n>]`**

Fetch size of an internal cursor.

**`[-all]`**

Evolve all classes. Class name is not necessary with this option.

**`[-threads <n>]`**

Fork N threads for evolve.

**`[-noprogess]`**

Do not show progress on the console.

**`[-verbose]`**

Show additional information, useful for debugging.

If you are not sure which of the schema's have multiple versions, you can use the tool in the following way:

```
dbtool -schevol -evol -all <database name>
```

This tool is designed in such a way that it evolves only those instances that need to be evolved. Hence, for a large database, using "-all" instead of specific classnames is absolutely fine.

A few things to note, before using the tool:

- This tool should be used only on an offline database. There should be no active transactions running during the operation of this tool.
- Since this is an offline tool, it requires the database to be in "DBA-only multi-connection mode", which can be achieved through `dbinfo -d <dbname>`.
- This tool can take a couple of hours on large databases. Please avoid terminating the tool forcefully.
- This tool is transactionally safe, therefore, incase of interruption, database consistency is guaranteed. Since the functionality of this tool is running in 1p mode, interrupting this operation may lead to instability of the database server.
- The tool causes a heavy load on the disk hosting the database data volumes and the log files. This is expected behavior.

- This tool needs a high number of page buffers during its operation. Hence, if it reports the Versant error, “E2996 SM\_E\_OUTOF\_CACHE\_MEMORY: No BufferPool Mem Available”, you should increase the `max_page_buffs` parameter in the server process profile and restart the operation.

## dbuser

```
dbuser cmd_alternatives [option] dbname
```

This utility is used to add, delete, change password or list users of a database `dbname`. You can also change the DBA password.

For a remote database, append the node name to the database name using the syntax `database@node`.

Only the Database Administrator (DBA) of a database has the privilege of adding/deleting users to/from the users list.

Substitute one of the following for `cmd_alternatives` and `options`:

### **-list**

This is the default option.

Lists users on the database access list.

### **-add**

Adds a user to the database access list. You must also supply a `-n` or `-P` option.

### **-delete**

Deletes a user from the database access list. You must also supply a `-n` or `-P` option.

### **-chpasswd**

Changes the password for a user. You must also supply the `'-n'` option.

The `'-opasswd'` and `'-passwd'` options may be specified while using in scripts.

The DBA password can be changed in the following manner if user is logged in as the DBA:

```
dbuser -chpasswd -n <DBAname> -passwd <newpassword> dbname
```

If you use `-add` or `-delete`, you must substitute one of the following for `option`:



---

**-n username**

User name.

An error is raised if the given username is not found in the database when you are doing a delete.

**-P**

Changes public access.

When public access is **ON**, all database users can access the database. When public access is **OFF**, only users in the user list can access the database.

To turn public access **ON**, use **-add -P**.

To turn public access **OFF**, use **-delete -P**.

If you use **-add** you may also use the following option:

**-m access\_mode**

Sets database access mode for the user(s) specified with **-n** or **-p**.

Alternatives for **access\_mode** are:

**r** Read-only access.

**rw** Read/write access, which is the default.

To see the current access mode of each user, use **dbuser -list**.

If you use **-chpasswd** you may also use the following options:

**-passwd password**

Specifies new password for the user.

**-opasswd old-password**

Specifies old/existing password of the user.

**-noprint**

Suppresses display message.

**dbname**

The database name.

To add a user to a database, you must be the owner of the database and the database must be a group database. To delete a user from a database, you must own the database. You cannot delete

the owner of the database. To perform any of the administrative activities on a database associated with a password-based authentication, the user has to provide the DBA password.

Only the database owner (DBA) can change the password of any other user, without specifying the old password. All other users need to specify the old password for verification, while changing their password.

If you have access to a database but are not the owner, you cannot give or delete the access privileges of others.

After a database has been created, you can change its ownership by changing the ownership of its database directory.

### Example:

By running dbuser utility, we are adding a user 'Fred' in the database named 'EmployeeDB'.

```
dbuser -add -n Fred -m rw EmployeeDB
```

```
Enter Fred's password:
```

```
Confirm Fred's password:
```

- To make the database `publicDB` available for public access

```
dbuser -add -P publicDB
```

- To delete a user Allen from the database `EmployeeDB`

```
dbuser -delete -n Allen EmployeeDB
```

- To list all the user information in the user list of the database `EmployeeDB`

```
dbuser -list EmployeeDB
```

- A database user can change the password associated with the user information.

For example, Fred's password can be changed by running the following:

```
dbuser -chpasswd -n Fred EmployeeDB
```

```
Enter Old Password :
```

```
Enter New Password :
```

```
Re-Type New Password :
```

---

## dropclass

### dropcls

```
UNIX dropclass options [-d dbname] class1 [class2 class3 ...]
WIN dropcls options [-d dbname] class1 [class2 class3 ...]
```

This utility drops one or more specified classes, their subclasses and their instances from the specified database `dbname`.

For a remote database, append the node name to the database name using the syntax `database@node`.

**WARNING:-** This utility is used only in special circumstances since the Schema Change Utility (`schcomp`) can also drop classes when a class must be replaced.

#### Parameters for this Utility are:

##### **options**

Drops Class options.

##### **-d dbname**

The name of the database from which a class and its instances are to be dropped. The default is the database named in the environment variable `O_DBNAME`.

For a remote database, append the node name to the database name using the syntax `database@node`.

##### **class1 ...**

A list of classes to be dropped.

#### Options for this Utility are:

The default option is:

##### **-n**

Makes no changes to the database but reports what classes would have been dropped if the `-y` option had been given.

##### **-p**

Password corresponding to the user name entered in `-u`. This is mainly intended for scripts so that users don't have to enter the password interactively.

For normal usage, users can just pass in the `-u` option and dropclass will prompt for the password. Passwords cannot be more than 256 characters in length. Running with just the `-p` option without the `-u` option will return an error.

**-u**

User name to access the DB. This user has to be a valid user of the DB. You will be prompted for a password if no password is entered using the `-p` option. Username cannot be more than 32 characters in length. The default is to run as the OS user without any password.

**-y**

Drops the specified classes and their instances without asking for approval.

**-i**

Interactive <default>: Ask Yes or No. (`stdin` must be `tty`)

## dropinst

```
dropinst options [-d dbname] class1 [class2 class3 ...]
```

This utility removes all instances of one or more specified classes and their subclasses from the specified database `dbname` without changing the database schema.

For a remote database, append the node name to the database name using the syntax `database@node`.

### Parameters for this Utility are:

#### options

Dropinst options.

**-d dbname**

The name of the database from which the instances are to be dropped. The default is the database named in the environment variable `O_DBNAME`.

**class1 ...**

---

A list of classes whose instances are to be dropped.

**Options for this Utility are:**

**-p**

Password corresponding to the user name entered in `-u`. This is mainly intended for scripts so that users don't have to enter the password interactively.

For normal usage, users can just pass in the `-u` option and `dropinst` will prompt for the password. Passwords cannot be more than 256 characters in length. Running with just the `-p` option without the `-u` option will return an error.

**-u**

User name to access the db. This user has to be a valid user of the db. You will be prompted for a password if no password is entered using the `-p` option.

Username cannot be more than 32 characters in length. The default is to run as the OS user without any password.

**-y**

Drops the instances without asking for approval.

Instances created by any language interface can be dropped.

For example, to drop all instances of `PObject` and all classes deriving from `PObject` from the database `myDB` without changing the database schema:

```
dropinst -y -d myDB PObject
```

To do the same thing as another valid user of the DB other than the current OS user you are logged in as:

```
dropinst -y -d myDB -u user1 -p luser1 PObject
```

or

```
dropinst -y -d myDB -u user1 PObject
```

This will prompt you for a password.

**-i**

Interactive <default>: Ask Yes or No. (`stdin` must be `tty`)

**-n**

Makes no changes to the database but report what instances would have been dropped if the -y option had been given.

## ftstool

`ftstool option dbname`

This utility is relevant only if you are using the Fault Tolerant Server option (synchronous database replication).

The purpose of Vedding (FTS) is to keep two databases in synchronization so that the applications can continue operating even if one of the databases fails.

The `ftstool` utility can be used on the Fault Tolerant Database:

- To control the automatic synchronization of the Fault Tolerant Server, options:  
-pollingdelay (or -p) -1, 0.
- To get the database and polling state of the FTS database and
- To force a database fail-over for administration purpose, whenever necessary. The options for this are -disable and -forcedisable.

By default, if one database in a replication pair goes down, the other database starts a polling process to watch for its return. Then, when the failed database returns, the polling process automatically re synchronizes the DOWN database with the UP database. After the synchronization is complete, the polling process removes itself from memory.

**For more information, please refer to the *Versant Vedding Usage Guide*.**

Once you have invoked `ftstool`, the ON or OFF setting for automatic synchronization is persistent.

For example - if you turn automatic synchronization OFF and then stop the database, even when the database restarts, automatic synchronization will still be OFF.

**Parameter for this Utility are:**

**dbname**

Name of database.

## **option**

Synchronization option.

### **Options for this Utility are:**

The default option is :

**-pollingdelay (sub-options)(or -p (sub-options))**

This option is used for the polling process

The sub-options for this utility (-p) are :

**0**

Immediately starts a polling process, once any database server goes down.

**-1**

Disables the polling process.

It does not shut down the replica database. Replication continues, but once any database server goes down, polling and synchronization do not occur.

**NOTE:-** Values other than 0 and -1 are not valid.

**-stopsync**

Stops polling process and discards accumulated polling records.

This option is used while performing an online replica rebuild with the help of HABackup or vbackup utility.

**-disable**

Use this to force a fail-over manually.

Disables polling process on its replica database, waits for the active transactions to complete and then stops the database. In case of long running transactions this option may not be advisable as it waits for the active transactions to complete and instead **-forcedisable** option can be used.

**-enable**

End the manual fail-over.

This will restart the DOWN database and start the polling process on the replica database so that the DOWN database is synchronized with its replica.

## **-forcedisable**

Same as `-disable`, but does not wait for any transaction to complete.

Disables polling process on its replica database, waits for the active updates to finish and then stops the database.

## **-status**

Displays the state of the FTS database and its polling process

An FTS database can be in any one of the following FTS states:

FTS State	Description
Default/UP	The database is running normally and in sync with its replica pair.
Unstartable/DOWN	The database is in the recovery mode and is waiting for the polling process to complete synchronization before allowing any connections.
Polling	The polling process has started from the database and is trying to connect to its DOWN replica database. Until it is not able to successfully connect, the state of the UP database will be POLLING and all the resynchronization records will be saved.
Syncing	The polling process has succeeded in connecting to the DOWN database and is performing resynchronization.
Syncing Done	Polling has completed synchronization process.
Suspend	This implies that the <code>ftstool -stopsync</code> has been applied on the database and as a result of which no resynchronization records are being saved.
Awaiting Restore	Database backup with <code>-startsync</code> option has been completed and resync records are getting accumulated.

The polling states are:

Polling State	Description
Enabled	Polling process is activated and will auto start if its replica database goes DOWN.



---

Disabled	Polling process is deactivated and will not start if its replica database goes DOWN.
Requested	Polling process has been requested.
Forked	Polling process has started from the UP database.

**NOTE:-** If the database is authenticated using the password mechanism, then the DBA must ensure that the same password is specified for the replica pair, else `SM_E_INVALID_PASSWORD` error may be thrown.

## Typical sequence of events

The purpose of the Vedding (FTS) is to keep two databases in synchronization even if one of the databases fails.

Following is a sequence of events that occur when one of the replica databases goes DOWN.

1. Applications continue to work normally, but the resynchronization records will be saved on the UP database.
2. Polling automatically begins, and the polling process changes the state of the UP database to `POLLING` and tries to connect to the DOWN database.
3. When the DOWN database returns, the polling process from the UP Database connects to the DOWN database and starts the resynchronization process. At the beginning of the resynchronization, the polling process changes the state of the UP database from `POLLING` to `SYNCING` and once the resynchronization is complete, the database state is changed to `SYNCING DONE`.  
From now onwards, the new client connections will connect to both the replica databases.

## Restoring a replica database

Suppose the failed database has been destroyed or suppose that polling is taking too long to resynchronize the two databases, in such cases you might want to do the following:

### 1. Stop synchronization

Use `ftstool` utility to stop synchronization:

```
ftstool -stopsync dbname
```

This will change the status of the operating (UP) database from "POLLING" or "SYNCING" to "SUSPEND" and discard accumulated re synchronization records.

If the specified database is in any state other than "POLLING" or `SYNCING` when this option is used, the error `UT_DB_WRONG_STATE` will be returned.

If the failed database comes back on line after this option has been used, no synchronization will occur.

## 2. Fix the failed database

Or use the `makedb` utility to create a new database to be used for replication.

If you create a new database, its name should be the same as the failed database.

For example, suppose that the failed database is totally destroyed.

1. Remove the failed database and its directories:

```
removedb -rmdir dbfailed
```

2. Recreate the directories and database files, using the same name as the database that has failed:

```
makedb dbfailed
```

## 3. Back up the running database using `vbackup -startsync` option

For example, to backup `dbname` to a file named `dbname.bak`:

```
vbackup -dev dbname.0 -startsync -backup dbname
```

**See also “vbackup” on page 222.**

If you use the `-startsync` option and the specified database name is not found in the FTS replica file, an appropriate error will be returned.

Using the above options, the database changes its state from `SUSPEND` to `AWAITING_RESTORE`.

However, if the status of the running database is not `SUSPEND` when the `vbackup` command is executed, then the error `UT_DB_WRONG_STATE` is thrown.

If an error occurs during backup process, step 1 (`ftstool -stopsync`) has to be replayed.

## 4. Restore the database

By using the backed up file, generated in step 3 by using the `vbackup -restore -rename` option

For example,

```
vbackup -dev dbname.bak -restore dbname -rename dbfailed
```

The above command will automatically start polling and resynchronize the `dbfailed` database, when the restore process has been completed.

**NOTE:-** Unless automatic polling has been turned off, this process is not successful.

**See also “vbackup” on page 222.**

---

## makedb

makedb [options] dbname

This utility will create a database directory named `dbname` as per the options in the options parameter and create the database support files, if they do not exist.

**NOTE:-** Before creating a database with `createdb`, you must run `makedb` or create a database directory and the profile files manually.

For a remote database, append the node name to the database name using the syntax `dbname@node`.

### Options for this Utility are:

**[-g | -p]**

**-g**

This is the default option which is for a Group database.

Group databases are accessible to many users at the same time.

**-p**

Personal database.

Personal databases are accessible to one application at a time.

**-owner user**

Makes user the owner of the database directory. The default is to use the current login name.

You can make someone else the owner only if you are logged in as the superuser.

**-cpprofile db**

Copies the profile file `profile.be` from the specified `db` directory to the directory for the new database.

A user can copy a database profile only if it is the owner of the database i.e., only the DBA is allowed to copy profiles.

This option works only if both databases are on the same machine.

If only a database name is used rather than a full path name, the `db` directory under the database root directory is searched.

**CAUTION:-** While using this option, great care must be taken to ensure that the volume paths for system volume (`sysvol` parameter), log volumes (`physical.log` and `logical.log` parameters) and data volumes (`datavol` parameter) in the source server process profile file (the `profile.be` file that is being used as a reference to create the new `profile.be`) should not have absolute paths.

If an absolute path is specified then please make sure that those paths are not referenced by any other database's server process profile file (`profile.be`) that is on the same machine on which the target database directory will be created. Failure to check this can result in severe database corruption and / or deleted volumes.

**-nofeprofile**

Does not create an application process profile.

**-logging**

Turns transaction logging `ON`.

**-locking**

Turn transaction locking `ON`.

**-noprint**

Does not display any messages while the command runs.

**-sglatch**

Turns database multi-latching `OFF`.

By default, the database server process will use multi-latching.

**See also “multi\_latch” on page 114 in “Chapter 3 - Database Profiles”.**

**-promptpasswd**

Associates a password with DBA to enforce DBA authentication.

The length of the password cannot exceed 255 characters. If string `"null"` is specified as password value, then the DBA will not be associated with a password.

**-beprofile *file***

Copies the said file as a server profile file.

If the 'file' is not specified with absolute path, it is searched in the current working directory.

This parameter cannot be specified with `-cpprofile`.

**NOTE:-** In case of 2p models, `makedb` will search for the file at the specified location on the same machine, where the database directory for that database is located. If the file does not exist or is not accessible, `makedb` will fail with error `UT_ER_MAKEDB_PROFILE_NOTFOUND`. The file path has to be an absolute path.

**CAUTION:-** Please read the “Caution” given for `-cpprofile` option. Similar care should be taken while using this option too.

#### **`-feprofile file`**

Copies the said file as a application profile file.

If the <file> is not specified with absolute path, it is searched in the current working directory.

This parameter cannot be specified with `-nofeprofile` and `-cpprofile`.

**NOTE:-** In case of 2p models, `makedb` will search for the file at the specified location on the same machine, where the database directory for that database is located. If the file does not exist or is not accessible, `makedb` will fail with error `UT_ER_MAKEDB_PROFILE_NOTFOUND`. The file path has to be an absolute path.

The directory created will be:

#### **`/dbroot/dbname`**

A directory for the database that branches from the database root directory. The name of the directory will be the same as the database name specified as `dbname`.

Database support files that will be created, if they do not already exist, are:

#### **`../.osc/dbname`**

An application process profile file, located in the `.osc` directory branching from your home directory. This file contains operating parameters used when this database is a session database.

The operating environment for the application will be set per the specifications in the application profile file corresponding to the session database. If no application profile is found, the application will use default environment settings.

#### **`profile.be`**

A database server process profile file, located in the database directory. This file contains database creation and operating parameters.

## **.lock**

A lock file, which indicates whether or not the database has already been started.

## **.pw**

A password file is created only if the DBA authentication is done using a password-based mechanism. The specified password is stored in this file and is accessed every time a utility that requires DBA authentication is invoked like `createdb`, `dbinfo`, `vmovedb` etc. If DBA forgets the password, the DBA can log as OS user into the machine, delete the password file and recreate a new password using the `dbuser` utility.

**See also “dbuser” on page 182.**

**CAUTION :-** The DBA authentication will not be performed in the absence of this file under any circumstances.

## **.vbackup**

This file contains data necessary for reporting vbackup progress. Any attempts by the user to tamper with the `.vbackup` file will result in incorrect progress bar of vbackup.

## **.dbtype**

A database type file, which indicates whether this is a personal or group database.

For example:

Command	Result
<code>makedb db</code>	<code>db</code> is a group database
<code>makedb -g db</code>	<code>db</code> is a group database
<code>makedb -p db</code>	<code>db</code> is a personal database

## makeprofile

```
makeprofile [options] dbname
```

---

This utility makes application process and server process profiles for a database named `dbname`.

For a remote database, append the node name to the database name using the syntax `database@node`.

**Options for this Utility are:**

**-cpprofile db**

Copies profiles from the specified `db` directory.

A user can copy a database profile only if it is the owner of the database i.e., only the DBA is allowed to copy profiles.

If only a database name is used rather than a full path name, the `db` directory under the database root directory will be used.

**-nofeprofile**

Does not create an application process profile.

**-logging**

Turns logging `ON`.

**-locking**

Turn locking `ON`.

**-noprint**

Suppress display messages while command runs.

**-sglatch**

Turn database multi-latching `OFF`.

By default, the database server process will use multi-latching.

**For more information refer to “multi\_latch” on page 114, in “Chapter 3 - Database Profiles”.**

**-beprofile file**

Copies the said file as a server profile file.

If the file is not specified with absolute path, it is searched in the current working directory.

This parameter cannot be specified with `-cpprofile`.

**NOTE:-** In case of 2p models, `makedb` will search for the file at the specified location on the same machine, where the database directory for that database is located. If the file does not exist or is not accessible, `makedb` will fail with error `UT_ER_MAKEDB_PROFILE_NOTFOUND`. The file path has to be an absolute path.

## **-feprofile file**

Copies the said file as a server profile file.

If the file is not specified with absolute path, it is searched in the current working directory.

This parameter cannot be specified with `-nofeprofile` and `-cpprofile`.

**NOTE: -** In case of 2p models, `makedb` will search for the file at the specified location on the same machine, where the database directory for that database is located. If the file does not exist or is not accessible, `makedb` will fail with error `UT_ER_MAKEDB_PROFILE_NOTFOUND`. The file path has to be an absolute path.

For example, to make profiles for `myDB`:

```
makeprofile myDB
```

If you are recreating a database previously removed with the `removedb` utility, you can reuse the directories and profiles without having to run `makeprofile` or `makedb`.

## NetworkServices

### NetworkServices

On Windows machines, Versant provide a graphic interface that allows you to use multiple releases on the same machine.

You can use the `NetworkServices.exe` graphical user interface (GUI) to configure the Versant service, `versantd`, to listen and respond on multiple ports so that you can use multiple Versant releases on the same machine.

To perform the same task manually, you can set the `VERSANT_SERVICE_NAME` environment parameter (as described in the chapter "Configuration Parameters.")

If you have multiple Versant releases on your machine, you may want to connect to servers from one or the other of these releases selectively from different clients. This is an advanced option and requires careful configuration. This is not something you normally need to do, and we recommend that you do it only with the guidance of Versant Technical Support.



## About the Versant system service

The TCP/IP system service that listens for database requests is named `versantd` on Windows machines and `oscscsd` on Unix machines.

In a default installation, the Versant connection services daemon listens on TCP/IP port 5019. In the default case, all clients try to connect to the Versant service on the server machine, and the service name is mapped to a port number using the TCP/IP configuration file on the client machine (not the server machine).

By using the `NetworkServices` GUI, you can cause a client application to connect to a different service name and, therefore, a different port number. Correspondingly, on the server machine you can set up different Versant releases to listen on different port numbers and allow a client to connect to a Versant server process from the release of their choice.

During Versant installation on a Windows machine, `versantd` is added to the list of available services and automatically starts when Windows starts.

When `versantd` starts, it does the following:

1. It reads the services file located in the system directory, looking for service names that begin with the prefix "osc". The services file contains port numbers for well-known services.

The location of the services file is:

```
%SYSTEM_ROOT%\System32\drivers\etc\services
```

The services file has entries in the format:

```
<service name> <port number>/<protocol> [aliases...] [#<comment>]
```

For example:

```
ftp                21/tcp
telnet             23/tcp
smtp              25/tcp      mail
oscscsd           5019/tcp    # VERSANT connect service
```

2. The `versantd` daemon next retrieves information about the services from the registry.

The location it reads is:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services..
\Versantd\Services Config
```

At this location, `versantd` will find three key pieces of information:

- the Service Name
- the name and location of the process associated with the service

- the port number that the `versantd` will monitor which, in turn, will spawn the associated process.

Once `versantd` finishes its initialization phase, it monitors the appropriate ports and spawn the associated process once an action is requested on that port.

All significant actions associated with the `versantd` program are recorded in the Application Event log.

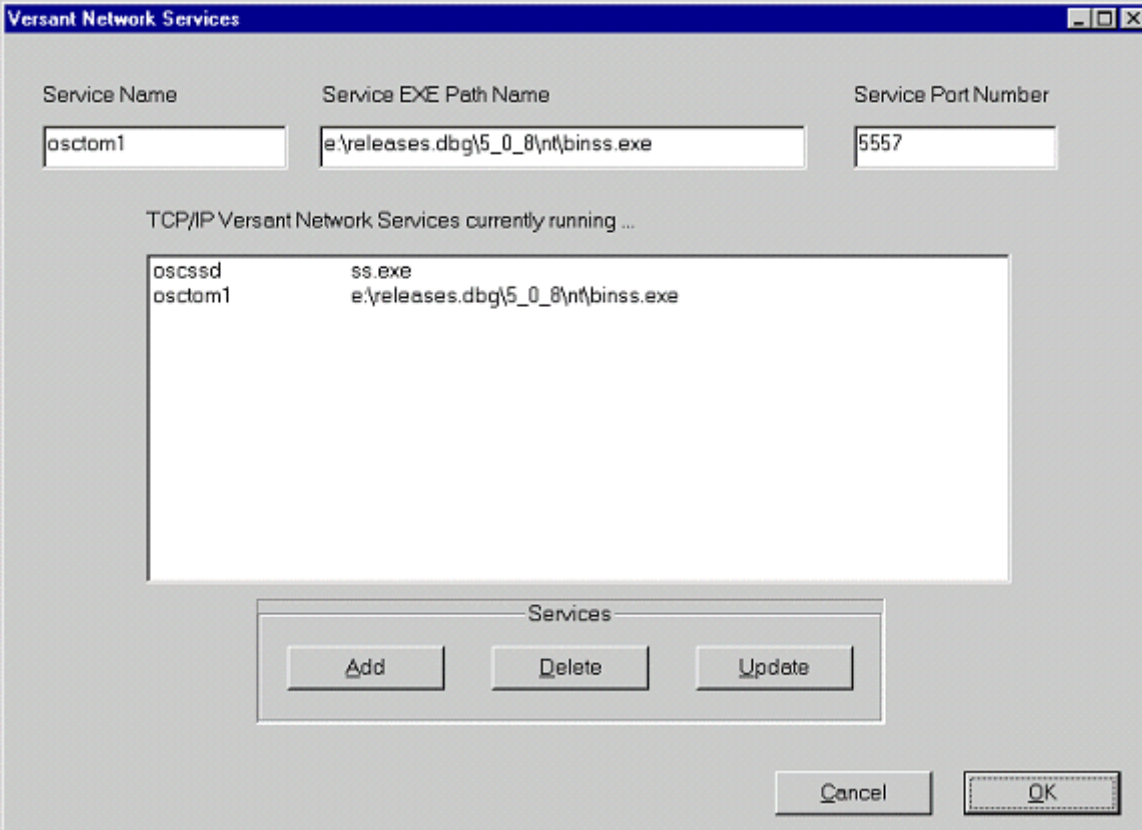
On Windows, you can monitor a maximum of 64 ports.

## Adding additional ports and services

If you want `versantd` to monitor additional ports, you can use the Versant Network Services interface.

To open the Versant Network Services interface, double-click on `NetworkServices.exe` in your `VERSANT \bin` directory.

The Versant Network Services interface looks like this (actual entries will vary depending on your configuration):



The image shows a Windows-style dialog box titled "Versant Network Services". It has three input fields at the top: "Service Name" with the text "osctom1", "Service EXE Path Name" with the text "e:\releases.dbg\5\_0\_8\nt\binss.exe", and "Service Port Number" with the text "5557". Below these fields is a text label "TCP/IP Versant Network Services currently running ...". Underneath this label is a list box containing two entries: "oscsd" with "ss.exe" and "osctom1" with "e:\releases.dbg\5\_0\_8\nt\binss.exe". At the bottom of the dialog is a "Services" section containing three buttons: "Add", "Delete", and "Update". In the bottom right corner of the dialog are "Cancel" and "OK" buttons.

Service Name	Service EXE Path Name	Service Port Number
osctom1	e:\releases.dbg\5_0_8\nt\binss.exe	5557

TCP/IP Versant Network Services currently running ...

Service Name	Service EXE Path Name
oscsd	ss.exe
osctom1	e:\releases.dbg\5_0_8\nt\binss.exe

Services

Add Delete Update

Cancel OK

In this interface, you can add a new service, delete a service or update an existing service.

### Add a service

To add a new service name, type a new name in the Service Name Box, enter either the default of `ss.exe`, or the full pathname of the program associated with the service, and the proper port number.

You must be careful to enter a valid port number. If you do not enter a proper port number, you will cause problems and conflicts on your system.

After making your entries, simply press the Add button, and the new service will be added.

## Delete a service

To delete a service, select it and then press the Delete button.

## Update a service

To update an existing service, select the appropriate service name and make the necessary changes in the edit fields desired.

After making your changes, press the Update button. The changes will be reflected in the list.

## Restrictions on service name

There are some restrictions associated with the service name.

- The service name begin with the prefix "osc".
- The service name can only contain the alphanumeric characters, a-z, A-Z, 0-9. It cannot contain a dash, underscore, or any other characters.

Use of invalid characters will cause problems. There is a check and a force for the prefix: if you do not have the `osc` prefix characters in the service name, the prefix `osc` will be automatically added.

## Record the changes

When you have finished making your changes, press the `OK` button to terminate the GUI and permanently record the changes.

To abort the changes, press `Cancel`.

When you press `OK`, the following will happen:

- The OK button will change to WAIT.
- The services configuration is saved in the registry in the following location  
`"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services.. \Versantd\Services Config"`
- The services file is updated.
- The TCP/IP service is stopped and restarted, which forces it to read the changes in the services file.

The `versantd` service is stopped and restarted, which forces it to read the changes in the services file and registry.

---

## oscp

`oscp parameter`

This utility displays information about your Versant environment for the release you are using.

**Options that you can substitute for `parameter` are:**

(with examples shown for a UNIX installation)

**-b**

Prints the patch information including Versant release and patch number, operating system supported, compiler versions. List the ODBMS and cxx bugs fixed in this patch.

Use `vinfo -r` to get a full list of bug fixes for all VOD components.

**-d**

Returns your Versant database root directory.

For example, the return information might be:

```
/versant/db
```

**-i**

Returns your Versant product version, software root path, runtime path, database root path, machine and directory containing the `osc-dbid` database system file, and `ss.d` location.

**-l [`@remote_host`]**

Returns the release level of this version of `oscp`. The default is the level of `oscp` on your local machine; to find the release level on another machine, specify the name of the remote machine using `@remote_host` syntax.

**-r**

Returns your Versant software root directory.

For example:

```
/usr/local/versant
```

**-p**

Returns your run time path (the software root plus the release and platform directories).

For example:

```
/usr/local/versant
```

**-v**

Returns your Versant version number.

**-n**

Prints the machine containing the `osc-dbid` database system file.

**-o**

Prints the directory containing the `osc-dbid` database system file.

## polling

```
polling fromdb
```

This utility manually synchronizes two databases in a replication pair, using the database specified as `fromdb` as a source.

The target database will be found in the Fault Tolerant Server configuration file.

This utility is useful if you are using synchronous database replication (the Fault Tolerant Server option) and have disabled automatic synchronization with the `ftstool` utility.

**For more information on synchronization, See also “ftstool” on page 188.**

**For information please refer to the *Versant Veddin Usage Guide*.**

## removedb

```
removedb [options] dbname
```

This utility stops the database if it is running, destroys and removes all volumes of the database `dbname`, and deletes the database from the system database identifier file `osc-dbid`.

For a remote database, append the node name to the database name using the syntax `data-base@node`.

---

**Options for this Utility are:****-f**

Forcibly stops and removes the database immediately even if it is currently running and other users are connected to it.

Without the `-f` option, the database will not be removed if it is in use by an active transaction. If it is in use, the message "Database in Use" will be printed and `removedb` will be terminated.

**-rmdir**

Removes the database directory and all files in it as well as the database.

**-noprint**

Suppresses display messages while running.

The utility gets user confirmation before starting the removal procedure. Key in "y" to continue with the removal of database or any other character or enter will be treated as "No". If you use `-f` option to remove the database, then the database will be removed without confirmation.

The database system `osc-dbid` file must be visible from your machine before you run `removedb`. Only the owner of a database can remove it. If the database is authenticated using the password-based mechanism, then the DBA password should be provided.

When you remove a database, all volumes associated with the database are removed. This includes any extra storage volumes that have been added after the database was created. However, the database directories, profiles and password file are not deleted when you run `removedb` unless you specify `-rmdir`. If you do not delete the directories and files, you can recreate a removed database using the same directories profiles and password if any. If you delete database directories manually but do not run `removedb`, the database numerical identifier and information about the old name and path remain in the `osc-dbid` database system file. Also, if you delete the database directories manually, shared memory will not be removed.

If you accidentally delete database files manually and now want to remove the database information from the `osc-dbid` file, use the command:

```
dbid -d mydb
```

**NOTE:-** It is recommended to use the above command only in the mentioned circumstances.

**Examples:**

```
removedb my_db
removedb -f my_db
removedb -rmdir mydb
```

## removereplica

### removrep

```
removereplica [options] replicadb
```

This utility stops synchronous database replication, removes the synchronous replication database named `replicadb`, and leaves the database directory and profile files intact. If the replica database is associated with a password during creation, then the utility requires the DBA password for authentication.

After calling `removereplica`, you must manually remove the entry for the synchronous replication database in the replica file for the Versant installation you are using,

Since the database directory and profile files are left intact, you do not have to rerun `makedb` before creating a new database with the same name either with `createdb` or `createreplica`.

Using `removereplica` is the same as using `removedb` and then manually editing the replica file with a text editor.

You can call `removereplica` from a program by using the C/Versant `o_removereplica()` function or the C++/Versant `removereplica()` method.

#### Options for this Utility are:

**-f**

The database must be stopped before it can be removed. The `-f` option will forcibly stop the database and is the same as using `stopdb -f`.

**-rmdir**

Removes the database directory and profile files as well as the database.

**-noprint**

Suppresses display messages while running.



---

## reorgdb

`reorgdb [options] dbname`

This utility reorganizes all data volumes for the database `dbname` by rearranging objects to remove gaps in physical space.

The purpose of the `reorgdb` utility is to pack data so that space is used more efficiently.

### Option for this Utility is:

#### **-noprint**

Suppresses display messages while running.

For a remote database, append the node name to the database name using the syntax `dbname@node`.

### Steps to reorganize a database

1. Set the database to single connection mode. For example:

`dbinfo -l mydb`

2. Stop the database.

For example : `stopdb mydb`

3. Run `reorgdb`.

For example : `reorgdb mydb`

## sch2db

`sch2db [options] schema.sch [schema2.sch ...]`

Load or change a class definition. This is the schema change utility.

### Options are:

#### **-D db\_name**

A specified personal or group database. The default database is the database specified in the environmental variable `O_DBNAME`.

#### **-e**

Attempt to evolve the schema to match class definitions in the schema file. In some cases, evolution will not be possible.

**-f**

Force the schema to match the specified schema files by dropping classes. Dropping a class definition means that data instances associated with the class are also dropped.

**-n**

Make no changes to the database. Report what changes need to be made to match the schema in the database with that in the .sch files.

**-p**

Password corresponding to the user name entered in `-u`. This is mainly intended for scripts so that users don't have to enter the password interactively. For normal usage, users can just pass in the `-u` option and `sch2db` will prompt them for the password. Passwords cannot be more than 256 characters in length. Running with just the `-p` option without the `-u` option will return an error.

**-q**

Force `sch2db` to work in silent mode.

In tandem with the `-y` option the "Ok to apply changes" question is inhibited.

With just the `-q` option, you will still have to confirm changes.

Error messages and warnings cannot be suppressed and will still be printed to standard error.

**-r**

Rename attributes rather than dropping and then replacing attributes.

**-u**

User name to access the DB. This user has to be a valid user of the DB. You'll be prompted for a password if no password is entered using the `-p` option. Username cannot be more than 32 characters in length. The default is to run as the OS user without any password.

**-y**

Do everything possible to make the database schema concur with the schema in the .sch files within possible restrictions set by `-e`, `-r` or `-f` options.

-?

Print Usage notes.

**For example:**

```
sch2db -d pdb1 -y -e schema1.sch schema2.sch
sch2db -D myDB example.sch
```

After all implementation files in a directory have been compiled using the Schema Capture Utility, you can insert all classes into the database with:

```
sch2db -D myDB *.sch
```

or

```
sch2db -D myDB -u user1 -p luser *.sh
```

If you want to run as a dbuser other than your current OS user login. You can also use the following format which will prompt you for a password.

```
sch2db -D myDB -u user1 *.sh
```

## Functionality

Whenever `sch2db` is invoked it connects to a specified database.

If the database does not have any predefined schema then the class definitions in the `.sch` (schema) files are loaded into the database.

If schema is being evolved, then `sch2db` compares the existing class definitions in the database with the corresponding class definition in the `.sch` file. Based on the comparison of all the classes, `sch2db` will provide you with a list of changes that will have to be made to make the schema in the database compliant with the schema in the `.sch` files. The changes are made once you approve the changes.

## Generating schema files

The input to the `sch2db` utility is always through `.sch` schema files created by using the `schcomp` utility. The input to the `schcomp` utility are schema implementation files that define the persistent classes to be created or changed.

**For example:**

If classes A and B in header files a.h and b.h need to be persistent in an application, then the schema.imp file would look like:

```
#include <cxxcls/pobject.h>
#include<a.h>
#include<b.h>
O_CAPTURE_SCHEMA(A); // tell the schema compiler that A is persistent
O_CAPTURE_SCHEMA(B); // tell the schema compiler that B is persistent
```

You must:

1. include all headers that are required to completely define classes A and B.
2. write an O\_CAPTURE\_SCHEMA statement for each persistent class.

Once you have the schemas, you can run the schema compiler (schcomp) on the .imp schema implementation files, which will generate the required .sch schema files.

The reference to the schcomp utility just above provides additional detail.

## Schema Evolution

Schema Evolution is the process wherein existing schema in the database is changed to reflect the changes in user classes.

All changes that are to be made to the database schema by sch2db can only be communicated to sch2db through the schema (.sch) files. Any time a user's class definition changes or a new persistent class is added to the user's header files, the schema files will have to be regenerated using the schema compiler.

Using schema evolution, the following changes can be made to the schema in the database:

- Add a new class to the database
- Drop an attribute from a class
- Add an attribute to a class.
- Completely redefine a class, by dropping the class definition in the database and adding the current one.
- Rearrange the attributes within a class.
- Rename attributes in a class.

**For more information, please refer to the Schema Evolution chapter in the *Versant Database Fundamentals Manual*.**

## Sch2db Usage notes

- When a class definition is changed, the logical object identifiers (loids) for instances of the class are not changed. This means that after class definition is changed there is no need to change application code or other objects that reference instances of the changed class.
- Changes made to class definitions are not applied to all instances all at once: changes are applied as instances are individually used in the normal course of later events. This avoids paralyzing a database by retrieving and altering all instances each time a class definition is changed.
- When schema is altered, the version history is maintained internally.
- The `-n` and `-y` options are mutually exclusive.
- Any changes to the existing schema (add, drop, etc.) will only be attempted if the `-e`, `-f` or `-r` options are used.
- The `-e`, `-f` and `-r` options are mutually exclusive and cannot be used in conjunction with one another. Any change required to synchronize the schemas must be approved by the user.
- You should evolve classes rather than creating new classes and copying data. If you modify a database schema by creating a new class, copying data, and dropping the old class, the copied objects will have new object identifiers. This means that links to the old objects are invalid.
- To ensure consistency of definitions, you should insert new class definitions in each database in which you will create or use instances of the class. This is recommended even though migrating an object will create class definitions in a new database if the class definition does not already exist.

Before running your application, `sch2db` helps in checking if the schema assumed by your application is the same as the schema in the database. Run `sch2db` with the `-e` and `-n` options providing all the requisite schema files. If the database schema matches with the application schema, then `sch2db` will not indicate any changes to be made to the database schema.

## schcomp

```
schcomp [c_options] [debug_options] file1.imp [file2.imp ... ]
```

Given an implementation file, create a schema description file and a schema definition file. This is the schema compile utility.

### Input

Input elements are:

## **c\_options**

Any option valid for your compiler's C preprocessor, such as `-I`, `-D`, and `-U`.

## **debug\_options**

Debugging options. These options are explained below.

## **file1.imp ...**

One or more C++ implementation files.

You cannot specify the list of implementation files with wild card characters.

## **Output**

For each implementation file, the following schema files will be created:

### **file1.sch ...**

A schema description file that describes classes in terms understandable to a database.

Schema description files are inserted into a database with the Schema Change Utility, `sch2db`.

### **file1.cxx ...**

A C++ schema implementation file.

For Versant, persistent capable classes, the `.cxx` file will contain class activation code.

## **Debugging Options**

Following are the options available for debugging with the schema compiler:

### **-preprocess**

Do preprocessing only. Write preprocessed text to the standard output.

### **-dependencies**

Do preprocessing only. Instead of the normal preprocessing output, generate on the preprocessing output file a list of dependency lines suitable for input to the UNIX make program.

### **-trace\_includes**

Do preprocessing only.

---

Instead of the normal preprocessing output, generate on the preprocessing output file a list of the names of files `#included`.

This is the same as the `-H` option of the Sun C++ compiler.

#### **-timing**

Generate compilation timing information.

This option causes the compiler to display the amount of CPU time and elapsed time used by each phase of the compilation (front end and back end) and a total for the entire compilation.

#### **-list list\_file**

Generate raw listing information in the file `list_file`. The information generated is useful for creating a formatted listing. The raw listing file contains raw source lines, information on transitions into and out of include files, and diagnostics generated by the front end.

Each line of the listing file begins with a key character that identifies the type of line, as follows:

**N:** A normal line of source. The rest of the line is the text of the line.

**x:** The expanded form of a normal line of source. The rest of the line is the text of the line. This line appears following the N line, and only if the line contains non-trivial modifications (comments are considered trivial modifications; macro expansions, line splices, and trigraphs are considered non-trivial modifications).

**s:** A line of source skipped by an `#if` or similar statement. The rest of the line is text. Note that the `#else`, `#elif`, or `#endif` that ends a skip is marked with an N.

**L:** An indication of a change in source position. The line has a format similar to the `#` line-identifying directive output by `cpp`.

#### **Example:**

```
L line-number "file-name" key
```

Where key is:

- for entry into an include file,
- for exit from an include file, and omitted otherwise.

The first line in the raw listing file is always an `L` line identifying the primary input file. `L` lines are also output for `#line` directives (key is omitted). `L` lines indicate the source position of the following source line in the raw listing file.

R for indications of a diagnostic remark: W for warning, E for error, and C for catastrophic error.

W indicates a diagnostic warning. The line has the form:

```
W "file-name" line-number column-number message-text
```

E indicates a diagnostic error. The line has the form:

```
E "file-name" line-number column-number message-text
```

C indicates a diagnostic catastrophic error. The line has the form:

```
C "file-name" line-number column-number message-text
```

Errors at the end of file indicate the last line of the primary source file and a column number of zero. Command-line errors are catastrophes with an empty file name ("") and a line and column number of zero. Internal errors are catastrophes with position information as usual, and message-text beginning with (internal error).

When a diagnostic displays a list (as an example, all the contending routines when there is ambiguity on an overloaded call), the initial diagnostic line is followed by one or more lines with the same overall format (code letter, file name, line number, column number, and message text), but lower case. The source position in such lines is the same as that in the corresponding initial line.

## **-xref**

Generate cross-reference information in the file `xfile`.

For each reference to an identifier in the source program, a line of the following form is written:

```
symbol-id name X file-name line-number column-number
```

where

symbol-id      A unique decimal number for the symbol.

X              An information code.



The information code *x* can have the following values:

D	A declaration that is not a definition.
M	Modification.
A	Address taken.
U	Used.
C	Changed, in the sense "used and modified in a single operation", such as an increment.
R	Any other kind of reference.
E	An error in which the kind of reference is indeterminate.

The fields of the above line are separated by tab characters.

## Embedding arrays

If an array of a class is embedded in another class, the C++/Versant run-time type identification code might call the default constructor for each element in the array. If this occurs and causes problems, insert the following expression in the constructor for the class of the array element:

```
class Address{
    Address(){
        if(O_DURING_SCAP) return;
        VPP_CLASS_CONSTRUCTOR1(Address);
        //user's code goes here...
    }
};
```

The above code is multi-thread safe. Without this, user would run into problem when he declares array of such objects in a persistent capable class, e.g.

```
class Person:public PObject{
    Address _addresses[2];
};
```

The value of `O_DURING_SCAP` will be set to non-zero values at run time for the purpose of schema capture.

For example:

```
schcomp -I'oscp -p'/h myschema.imp
```

The result will be the schema files myschema.sch and myschema.cxx.

## Backward Compatibility

The schema compiler for this release generates .cxx and .sch files that are NOT compatible with the ones generated by the schema compiler in releases prior to 5.2. This is mostly due to the fact that the current schema compiler doesn't generate in the schema files attributes that are internal to C++ (virtual table pointers and virtual base pointers.) The current schema compiler also doesn't generate "ghost" attributes (attributes that belong to redundant virtual base objects erroneously created by the cfront-based compilers in an object of the derived class) and other various kinds of paddings the old schema compiler generated in the schema files.

Also, the current schema compiler employs a different object layout algorithm from the old schema compiler and may generate a different class layout (the order of the attributes in a class may be different).

For example, for a class 'Person' the old and current schema compilers may generate different schemas as follows:

### OLD:

```
a 4 Person::age      0 !    ! ! o_4b          1 !      110 ! 0
a 8 Person::name     0 !    ! ! char         -1 !      86  ! 0
a 9 Person::__vptr    0 !    ! ! optr         1 !      84  ! 0
```

### NEW:

```
a 8 Person::name     0 !    ! ! char         -1 !      86  ! 0
a 4 Person::age      0 !    ! ! o_4b          1 !      110 ! 0
```

When schema evolution was performed using sch2db in previous releases of Versant, you would see the following response from sch2db:

```
Class 'Person' drop attribute 'Person::name'    Class
'Person' drop attribute 'Person::__vptr'
Class 'Person' gets new attribute 'Person::name'
Ok to apply changes?
```

If the answer was "yes", data stored in the database for attribute `Person::name` is lost.

The sch2db provided with the current release locates an attribute irrespective of the position it occupies and prevents the attribute from being dropped.

---

Redundant attributes are dropped. New ones are added and the schema for the class in the database is rearranged to match the order of the attributes in the schema file. The compatibility problem being solved, the old schema compiler is no longer required and hence is not a part of this release.

The current schema compiler generates schema files having version number equal to 116 (the first line in the `.sch` file provides the version number.) These schema files cannot be used as input to previous releases of the `sch2db` utility. However, the `sch2db` utility in this release is compatible with all schema files having version number 116 or lower.

## setdbid

```
setdbid dbid dbname
```

This utility sets the database identifier of a specified database to a specified value.

This utility goes to the machine containing the `osc-dbid` file, opens the `osc-dbid` file, and changes the entry associated with `dbname` to `dbid`. It also changes the value of the identifier in the database itself. It does not change the logical object identifiers of objects already defined in the database. If DBA authentication has been enforced on the database, then the user will be prompted for a DBA password.

### Parameters for this utility are:

**-noprint**

Suppress display messages.

**dbid**

New database identifier number.

**IMPORTANT:-** It is critical that the specified identifier must not have been used previously.

To find the current database identifiers of existing databases use `dblist` utility.

Also use `dbtool -AT -info -rootpages` to see all database identifiers currently used inside the database.

If you plan to copy, move or stream objects from other databases into this database you must perform the `dbtool -AT -info -rootpages` check on all these databases and ensure that none of those contains the new database identifier.

**dbname**

Name of the database.

## ss.d

`ss.d [command] [options]`

This utility adds a new command line option to the SS daemon to invoke it from the command line (does not require root privileges).

The new option accepts a port number on which the SS daemon listens for new client connections.

### Command:

#### Start

`-start`

Start the SS daemon on the specified port. Requires `-port` option.

### Options:

#### Port

`-port [port-number]`

A valid port number on the machine.

#### Log

`-log [log-filename]`

The valid filename (with complete path). To log more information from `ss.d` use this option.

### Example

Server

```
<DBA> ss.d -start -port 5080 /* start the SS daemon on port 5080 */
```

Client

---

```
<user1> set VERSANT_SERVICE_PORT=5080
<user1> db2tty -d _HYPERLINK
"mailto:testdb@<Server"<_Machine>
```

In the above example the connection is made to the SS daemon running on port 5080 on the server machine.

**NOTE:-** The command line invocation of ss.exe will work on Unix Platforms only.

## Usage notes

The SS daemon runs as the user who invokes it (with the new option) and can access the databases owned by that user. Typically a DBA should start the SS daemon before any connections are serviced. This is useful when embedding Versant ODBMS in other products.

## startdb

```
startdb dbname
```

This utility starts the database dbname.

For a remote database, append the node name to the database name using the syntax dbname@node.

**NOTE:-** Before starting a database that has password-based DBA authentication implemented, the user will have to provide the DBA password when prompted.

### Example:

```
startdb my_db
```

If the database was previously interrupted during a transaction, starting the database will automatically start a database recovery process.

For information on how to start a database without recovery from interrupted transactions, please call Versant Customer Support.

### Using the Versant database system typically involves the following basic actions:

1. Start personal and group databases that will be used by the application.
2. Start a session in a personal or group database.

3. Optionally connect to one or more group databases.
4. Disconnect from group databases. Disconnecting from a database will not stop it.
5. End a session. Ending a session will not stop the session database.
6. Optionally stop personal and group databases.

Starting a database with `startdb` creates an operating environment, performs any necessary recovery and cleanup operations and prepares the Versant Manager and Versant Servers for access.

Starting a database is optional, because an attempt to connect to a database will start that database if it is not already started. In the case of group databases, an explicit startup is preferred as it leaves the database ready for use without delay, and you have a chance to detect any possible errors as soon as possible.

To begin or end a database session and to make or break a connection to a group database, use language specific interface routines from within an application. To stop a database explicitly, use the `stopdb` utility.

After a crash, a `startdb` failure may not get reported on the screen and may have to be read from a file named `LOGFILE` in the database directory.

A `startdb` failure will also be indicated by the absence of the `cleanbe` process.

During execution of `startdb` you may see the following message:

```
Init SDA failed...
```

This means that the system does not have enough shared memory. Either increase system swap space, reduce the server process heap size, or stop some other database.

## stopdb

```
stopdb [options] dbname
```

This utility stops the database `dbname` and removes all database resources in memory.

For a remote database, append the node name to the database name using the syntax `database@node`.

**NOTE:-** Before stopping a database that has password-based DBA authentication implemented, the user will have to provide the DBA password when prompted.

---

To stop a database, you must be the user who started it, and thus the owner of its shared memory. After starting the `stopdb` operation, the user is prompted with a message to indicate the stopping of the database.

For example, to stop a database named `db_test`:

```
stopdb db_test
```

### Parameters for this Utility are:

#### **option**

Type of stop to perform.

By default, a database will not be stopped if it is in use by any application. The message "Database in Use" will be printed, and the `stopdb` utility will terminate.

The following options will override the default behavior.

### Options for this Utility are:

**[-f | -s | -st]**

#### **-st**

Wait for active transactions to complete and then safely stop the database. This option blocks new transactions.

If you attempt to start a new transaction while `stopdb` is waiting, the system returns the following error to the application : `SM_TR_XACTS_BLOCKED`.

Wait for active updates to finish and then safely stop the database. This option blocks both new updates and new transactions.

This option will wait for all active updates to finish before bringing the database down. If you try to start a new update while `stopdb` is waiting, the system returns the following error to the application : `SM_TR_NEW_UPDATES_BLOCKED`,

This option will also wait for all active transactions to finish.

#### **-f**

Immediately and forcibly stop the database.

Please use the `-f` option with care.

## **-noprnt**

Suppress display messages while `stopdb` is running.

## **vbackup**

`vbackup [options] command`

This utility manages database backup, restore and roll forward archiving operations.

The purpose of the `vbackup` utility is to allow recovery from a device failure or from accidental deletion of data.

Versant incremental backup (`vbackup`) strategy consists of three different backup levels viz. level 0, 1 and 2. As database size increases, it becomes more important to set the level in a way, which minimizes backup size and time. Level 0 performs a full backup. Level 1 backs up all changes made since the last level 0 backup. Level 2 backs all changes made since the last level 0 or level 1 backup, whichever was most recent.

Roll-forward (**RF**) archiving is another feature that preserves logical log records generated by a database during normal operation in a log archive. These records can be replayed on the database during recovery. Thus, RF archiving makes it possible to recover a database to its state just prior to the crash.

Thus, If you use just the backup and restore features, you can recover to the point of the last backup. But if you also use the roll forward features, you can recover to the point of the last committed transaction.

**NOTE:-** Please refer to the Versant GUI help for more details regarding the `vbackup` Progress-bar.

### **Command Parameters for this utility are:**

<b>Command</b>	<b>Description</b>
<code>-backup</code>	Backup a database.
<code>-restore</code>	Restore a database.
<code>-rename</code>	When used with <code>-restore</code> , restore to a renamed database.
<code>-overwrite</code>	When used with <code>-restore</code> and <code>-rename</code> , overwrite additional data volumes of restored database.



---

<code>-off</code>	Disable roll forwarding.
<code>-log</code>	Start roll forward archiving.
<code>-info</code>	Lists the set of backups and roll forward archives necessary to restore databases to their most recently backed up state.
<code>-list</code>	Lists the backup and roll forward files stored on a backup device or file. This parameter requires use of the <code>-device</code> option parameter.
<code>-resume</code>	When used with <code>-restore</code> , it resumes a suspended restore procedure.

## Options Parameter for this Utility are:

Options	Description
<code>-level</code>	Specify incremental level of backup.
<code>-device</code>	Specify the backup or restore device.
<code>-position</code>	Specify position on tape for backup or restore.
<code>-capacity</code>	Specify tape capacity.
<code>-blocking</code>	Specify number of bytes to read or write at a time.
<code>-comment</code>	Specify a comment to be associated with a backup.
<code>-rollforward</code>	Enable roll forward archiving during backup.
<code>-aggression</code>	Specify buffer flushing interval.
<code>-script</code>	Specify a script from which to run vbackup.
<code>-getbeprofile</code>	Extract the server process profile file <code>profile.be</code> from backup device or file. This parameter requires use of the <code>-list</code> command parameter.
<code>-odir</code>	Specify the directory location to extract the server process profile file <code>profile.be</code> . This parameter requires use of the <code>-list</code> command parameter.

`-startsync` When FTS is on and when used with `-backup`, the appropriate restore (the restore that restores the database that has been backed up with the `-startsync` option) will start the synchronization after the restore has completed.

## Vbackup usage notes

- To use `vbackup` on a database, you must be the DBA who created the database.
- To use `vbackup` on databases authenticated with password-based mechanism, you must specify the DBA password for all involved databases.
- If a database device is completely demolished, you can recreate the database directories on another device with `makedb` and then use `vbackup` to both create and restore the database in one step.
- If the recoverable errors "device not ready" or "end of tape" are encountered, you will be given a chance to correct the error and resume the backup or restore. Except for these recoverable errors, backups are not interactive and can therefore be called from a script.
- For options and commands, it is not necessary to specify the full flag as long as enough is specified to be unambiguous. For example, since no other flag begins with 'd', you can shorten `-device` to any of the following: `-devic` `-devi` `-dev` `-de` `-d`
- For `vbackup`, blocking option cannot be used with restore
- Estimating operating system resources: The `vbackup` utility will require an extra RAM of 2MB for `-backup` and `-restore` options to work properly.

**NOTE:-** The `multi_latch` parameter in the backend profile should be set to "ON" when backing up the database. If it is OFF, `vbackup` may hang.

## Command Parameters

The `command` parameters are explained in details as follows:

### (-backup) Backup database

`-backup [options] dbname1 [dbname2 ...]`

Backup one or more local or remote databases specified as `dbname`.

Use `dbname@host` syntax to specify a remote database.

## Usage Notes

You can backup multiple databases to a single tape drive or to a single file.

Database backups can be "online," which means that a database can be backed up while it is being used. Backups can be "incremental", which means saving only changes made since the last backup, or "full", which means saving the entire database.

If multiple databases are backed up with a single command, the first one will be stored in the position specified in the position parameter (see the option parameters below) and any additional ones will be appended after it.

A database backup saves the results of all transactions committed at the time `vbackup` is invoked. For additional safety, you may want also to use the `-rollforward` option in conjunction with the `-backup` command. This will ensure that no log records are discarded unless they have been archived, either with another backup or as a result of roll forward archiving with the `-log` command.

For example, to backup a database named "group" to the file `/tmp/level0` and also turn roll forwarding on:

```
vbackup -dev /tmp/level0 -rollforward -backup group
```

For this example, you will see the output as (the version number actually shown will match your release number):

```
VERSANT Utility VBACKUP Version 7.0.1.3
Copyright (c) 1989-2006 VERSANT Corporation
Backing up database 'group' to device '/tmp/level0':
      0%                50%                100%
      |                |                |                |
      .....
Backup has completed successfully.
```

## Backup database and start polling

```
vbackup -dev <backup device> [<options>] -startsync -backup <dbname>
```

After backing up the specified database, start a Fault Tolerant Server polling process to monitor the database.

If the specified database is not named in the FTS replica file, an error will be returned.

From now on, the specified database will be in the state `AWAITING_RESTORE`.

The restore process will then after it has been completed, synchronize the database, with the database that is up. All database transactions that happened after launching the `vbackup -backup -startsync` will then be in the database and hence both databases will be up and in sync again.

**See also “ftstool” on page 188.**

## (- restore) Restore database

```
-restore [options] dbname1 [dbname2 ...]
```

Restores one or more local databases specified as `dbname`.

Invoking `-restore` will immediately stop the specified databases (if running) and set them to `dba/single-connection mode`.

After the restore is complete, the database mode of the specified databases will be set to `multi-user mode`.

Before the restore begins, you are given an opportunity to save the current log files so that you can roll forward to the most recent committed transaction.

The first restore done must always be a level 0 restore.

Once this restore has successfully completed, you will be asked if you would like to do an additional level of restore or if you would like to quit. If you answer yes - to do another restore, you can insert a tape with any level 1 or 2 backup, that is incremental to the level 0 from which you just restored.

Before invoking the restore command, you must first locate the backups from which you will be restoring. If the database no longer exists, you must use your own records to determine the required backups. If the database is intact and you are restoring because you have accidentally deleted important data, you can use the `-info` command to list the backups necessary to restore to the most recently backed up state.

Once you know what your backups are, you must next locate the tapes on which they are stored. The `-list` command can be used to verify the presence and position of the backups on the tape.

If roll forward archiving was `ON` when the last backup was created, after you have restored from the backup tapes, you will be asked to insert the roll forward backup tape with the appropriate sequence number.

You will also be asked for a safe location to which the current database logical log file can be copied so that it is not overwritten during the restore.

You will be asked to insert successive roll forward backup tapes until you have no more backups to restore. At this point, type “quit” to exit.

## Failure conditions in Restore

When a restore starts, `vbackup` first copies the existing log file to a backup file name `_backup_.log`. If you encounter a failure during the restore operation, you need to copy the backup file to your logical log file, which is by default named `logical.log`.

After doing this, you can retry to restore the database. If you do not copy the backup file to your logical log file before repeating a restore, you may lose the changes made by all recent committed transactions.

For example, to restore the database “group” from the file `/tmp/level0:`

```
vbackup -dev /tmp/level0 -restore group
```

Following is a sample of the interactive dialog and output you will see during a restore.

```
VERSANT Utility VBACKUP Version 7.0.1.3
Copyright (c) 1989-2006 VERSANT Corporation
Restoring database 'group' from device '/tmp/level0':
During roll forward, would you like to apply records
from the database's current log file in addition to any
archived records ? [default = yes ]
This logical log file must then be copied to a safe
location so it is not overwritten during the restore.
Enter the path of the logical log [ default = '/versant/
db/group/logical.log' ]
Enter the path for storing the copy [ default =
'/versant/db/group/_backup_.log' ]
0%          50%          100%
|           |           |           |
.....
```

```
Restore has completed successfully.
Would you like to do another level of restore
on database 'group'? [ default = no ]
Current settings are:
    device    = '/tmp/level0'
    position  = 'current'
    capacity  = 'dynamic'
    blocking  = '10 Kilobytes'
Insert log archive #1 of database 'group'. [?=help] dev /tmp/log0
Current settings are:
    device    = '/tmp/log0'
    position  = 'current'
    capacity  = 'dynamic'
    blocking  = '10 Kilobytes'
Change additional settings or type <return> to proceed. [?=help]
Current settings are:
    device    = '/tmp/log0'
    position  = 'current'
    capacity  = 'dynamic'
    blocking  = '10 Kilobytes'
Insert log archive #2 of database 'group'. [?=help] quit
After you have verified that the database was
successfully restored, remember to remove
'/versant/db/group/_backup_.log'.
```

At this point when you quit, Versant will apply all log records in the file `_backup_.log` to the database.

If an error occurs and you cannot complete the restore and/or roll forward, before trying again, replace your logical log file with the `_backup_.log` file.

## (- rename) Restore and Rename database

```
-restore [options] dbname -rename dbnameNew
```

Restores the database specified as `dbname` to the database name specified as `dbnameNew`.

For example, suppose you have done the following:

1. Backed up a database named `db1` to a file named `db1.0`.
2. Used the `makedb` command to create database profile files and a database directory for new database named `db1new`.

---

The server process profile file (`profile.be`) can be edited to add absolute paths for volumes to be restored or can be copied from another database. In this case please make sure that the absolute paths used for the database volumes are not referred by any other database's server process profile file (`profile.be`) that is on the same machine on which the target database directory will be created. Failure to check this can result in severe database corruption and / or deleted volumes.

Now, suppose you invoke the following command:

```
vbackup -dev db1.0 -restore db1 -rename db1new
```

When you execute this command, a database named `db1new` will be created (using the profile file parameters as if you had invoked `createdb` manually) and initialized with the contents of the backed up database `db1`.

**NOTE:-** This command is especially useful if you are using the Fault Tolerant Server option and want to make an on-line recovery after a replica database has failed.

It is essential that the restored, renamed database has a different database identifier than the original database. Accordingly, if an entry for the new database name is found in the `osc-dbid` system file and the entry has the same identifier as the original database, an error will be returned.

To ensure that the renamed database has a unique identifier, you can delete any existing entry with the command:

```
dbid -d db1new
```

**See also “ftstool” on page 188.**

### **Usage Restriction:**

The restore rename takes only one database at a time.

### **(- overwrite) Restore, rename and overwrite**

```
-restore [options] dbname -rename dbnameNew -overwrite
```

Restores the database specified as `dbname` to the new database specified as `dbnameNew` and overwrite existing database volumes of restored database, with corresponding volumes of rename database.

For example, suppose you have done the following:

1. Backed up a database containing additional database volume at location `$TMP/voll` to device `dev1`.
2. Used the `makedb` command to create database profile files and a database directory for new database named `dblnew`.

Now, suppose you invoke the following command:

```
vbackup -dev db1.0 -restore db1 -rename dblnew
```

For this restore process, the following Versant Error Message will be raised:

```
UT_ER_DATAVOL_ALREADY_EXISTS
```

The user will be cautioned by `vbackup` restore command and prevent overwriting of additional data volume at location `$TMP/voll`.

Now, suppose you invoke the following command:

```
vbackup -dev db1.0 -restore db1 -rename dblnew -overwrite
```

Vbackup restore command is issued to overwrite data volumes of restored database with those of renamed database.

**NOTE:-** If the restored database has volumes at absolute paths, then please make sure that those paths are not referred by any other database's server process profile file (`profile.be`) that is on the same machine on which the target database directory will be created. Failure to check this can result in severe database corruption and / or deleted volumes.

### Usage Restriction:

Only one database name can be specified as an option to rename if the `-overwrite` option is to be used.

If more than one database names need to be specified, `-overwrite` will need to be specified explicitly for each `vbackup` command.

For Example:

```
vbackup -dev db2.0 -restore db2 -rename db2new -overwrite  
vbackup -dev db3.0 -restore db3 -rename db3new -overwrite  
vbackup -dev db4.0 -restore db4 -rename db4new -overwrite
```



---

## Warm standby (Incremental Restore)

The Warm Standby i.e., Incremental restore - strengthens Versant's high availability capabilities.

This feature is used as an incremental roll forward recovery. It is designed to minimize the down-time in an emergency event, which requires a database recovery.

To achieve this, an up-to-date copy of the primary database needs to be maintained - this is the Warm Standby database. In case of an emergency event, this Warm Standby database can be updated very quickly to the state of the primary database - just by applying the last roll forward archive plus the logical.log of the primary database (Rather than starting a full database restore that may take a considerable time with big databases).

### Features:

The purpose of Incremental Restore is:

- Whenever the restore process requests a new roll forward archive file, the restore process can be interrupted by a suspend (rather than a quit).
- Whenever the restore process gets suspended, the restore process can be resumed again by applying the next available roll forward archive files and/or the logical.log file.
- Between suspend and resume process, the database is in "Restore suspended (unstartable) mode", i.e., only the vbackup -resume is allowed to start and access this database.
- In case of an emergency the restore process needs only to be resumed with the last roll forward archive file(s) and/or the logical.log of the primary database in order to have the last transactions recovered in the Warm Standby database that is now ready to be used as the new primary database.

For this utility a separate license is necessary.

**For more information refer to "Versant Warm Standby Usage Guide".**

## (-log) Start Roll Forward archiving

```
-log dbname1 [dbname2 ...]
```

Start writing log records for the specified databases to the roll forward archive file.

Roll forward logging requires its own, ongoing process. This means that the command line will not return after this command is invoked until <Enter> is pressed a second time. Pressing <Enter> a second time will exit the logging process, which has the effect of suspending roll forward logging, which means that log entries will continue to accumulate changes without being written to archive files.

While running, the roll forward process will report any errors, which occur while writing to the archive tape or file. If the tape fills up, logging will be suspended, and you will be asked to insert a second tape.

You do not have to backup with the `-backup` command and the `-rollforward` option before using the `-log` command, although that is the recommended sequence. You can backup with the roll forward option and start roll forward archiving in either order, but you must do both in order to ensure that all database changes are archived and that you do not overflow your log files.

You can use the `-log` command to start roll forward archiving on a remote database. To specify a remote database, use the syntax `dbname@node`.

For example, to start writing log records for a database named “group” to the file `/tmp/log0`:

```
vbackup -dev /tmp/log0 -log group
```

For this example, the following output will be seen (the version number actually shown will match your release number):

```
VERSANT Utility VBACKUP Version 7.0.1.3
Copyright (c) 1989-2006 VERSANT Corporation
Press <return> when you are ready to exit.
Archiving log records to device '/tmp/log0':
Database 'group' is now being archived.
```

## (-off) Stop Roll Forward archiving

```
-off dbname1 [dbname2 ...]
```

Turns roll forward archiving off for the specified databases, which means that log records will not be written to the roll forward archive and may be discarded when the log file space is reused.

By turning roll forward archiving off, the roll forward archive chain gets interrupted, i.e. at restore time you can restore the database only up to the point when you turned roll forward archiving off.

To turn roll forward archiving on, perform a backup with the `-rollforward` option.

Only together with that backup the archive files can restore the database up to the last transaction (or up to the point when roll forward has been switched off).

---

## (-info, -list) Backup Info

### - info

```
-info dbname1 [dbname2 ...]
```

Shows information about the last backup that has been made for that database and about the log archiver state, whether a roll forward archiver is running or not. If one is running the roll forward archiver, it will show the number of the currently running archive.

```
Backup information for database `test1':
```

```
Level 0 backup made on `Tue May 23 17:11:46 2006'.
```

```
Currently archiving logs to file #57 via connection 289.
```

```
Done
```

### - list

```
vbackup -dev <device> -list
```

List the backup and roll forward files stored on a backup device or file.

Before restoring from a tape, it is common to use this command to list the contents of the tape or tapes from which you plan on doing the restore. This allows you to verify your records and to be sure that the backup files you are about to use are really on the tapes and in the positions where you believe they are.

List entries have the following format:

```
position <position>: 'label' volume <volume> blocking <blocking>  
'comment'
```

Where:

#### **position**

The position of the file on the tape, where 0 is the first file.

#### **label**

The label of the file.

**volume**

The volume number for the case where a backup spans multiple tapes. The first volume is 0.

**blocking**

The blocking factor that was used during backup on the device or file.

**comment**

The comment, if any that was entered when the backup was made.

**The format of a backup file label is:**

```
db_name -- level backup_level backup of date-time
```

Where:

**db\_name**

The name of the backed up database.

**backup\_level**

The level of the backup.

**date-time**

The date and time the backup was made.

**The format of a roll forward archive label is:**

```
Logs from date-time: db1 #file-no1 [ db2 #file-no2 ...]
```

Where:

**date-time**

The date and time archiving started.

**db1 ..**

The database that is being monitored.

**file-no ..**

The roll forward archive file sequence for the corresponding database.

---

**Examples:**

- `vbackup -dev /backup/test1.bak -list`  
VERSANT Utility VBACKUP Version 7.0.1.3  
Copyright (c) 1989-2006 VERSANT Corporation  
Listing Versant files on device `/backup/test1.bak':  
position 0: `level 0 backup of test1' volume 0 blocking `10 Kilo-  
bytes'.  
Done.
- `vbackup -dev /backup/test1.rf1 -list`  
VERSANT Utility VBACKUP Version 7.0.1.3  
Copyright (c) 1989-2006 VERSANT Corporation  
Listing Versant files on device `/backup/test1.rf1':  
position 0: `Logs from Tue May 23 17:16:29 2006: test1 #58' volume 0  
blocking `10 Kilobytes'.  
Done.

## Options Parameters

The `options` parameters are explained in detail as follows:

You can substitute zero or more of the following options.

### (- level) Backup level

`-level [ 0 | 1 | 2 ]`

The incremental level of backup to perform, either 0, 1, or 2. The default value is 0.

This option will be ignored unless it is being used with the `-backup` command.

The level options mean:

The default option is:

**level 0**

Perform a full backup.

**level 1**

Back up all changes made since the last level 0 backup.

## **level 2**

Back up all changes made since the last level 0 or level 1 backup, whichever was most recent.

The most recent set of backups can be viewed with the `-info` command. When a database is small, there is no problem with saving its entire state every time it is backed up, but as database size increases, it becomes more important to set the level in a way, which minimizes backup size and time.

## **(- device) Backup device**

`-device device_name`

The backup device on which to read or write.

The backup device can be either a tape drive or a normal file. If this argument is not set, it will default to the value of the TAPE environment variable.

To specify a remote device, use the syntax `host:device`. A remote device will be accessed using the `rsh` and `rmt` programs. Because `rmt` is used, remote devices are supported, but not remote files.

If you backup to a tape, you should use only non-rewinding tape devices.

On Windows, the default tape device name for SCSI tapes is `\\.\TAPE0`. See the notes for the Windows utility `vtape` for more information on device names.

## **(- position) Backup position**

`-position tape_position`

The position on the tape where the backup file should be read or written. The default is the current position.

### **Specify tape\_position as:**

`number`

A non-negative number. The first position on the tape is 0, the second is 1, and so on. If you are using a file for the backup, the only valid position is 0.

---

`current`

Use the keyword `current` to use the current position of the tape.

`append`

Use the keyword `append` to write after the last file on the tape.

Different tape drives have different capacities. If you want to put two backups on the same tape and your tape drive cannot do random positioning, then use `append` for the second backup on the same tape.

Versant uses standard operating system input/output control for positioning tapes, but individual device drivers may vary in their support. Positioning to the beginning of a tape or appending to the end are the most universally supported operations.

See the *Release Notes* for your platform, which may describe limitations on tape positioning.

## (- capacity) Backup Tape Capacity

`-capacity tape_capacity`

The maximum number of bytes to write to the tape before requesting a second volume.

The purpose of this parameter is to ensure that the backup does not attempt to write past the end of the tape.

**Specify `tape_capacity` as:**

**`number`**

A number with memory units.

The default unit is megabytes; alternately, you can specify by bytes, blocks, kilobytes, megabytes, or gigabytes. Units can be abbreviated. For example, `-cap 1.2g` means 1.2 gigabytes.

**`dynamic`**

This is the default option.

Use the keyword `dynamic` to trust the device to detect the end of the tape without losing any information.

Whatever capacity was used when making a backup must also be used when restoring from that backup.

## (- blocking) Backup Blocking

`-blocking blocking_param`

The number of bytes to read or write at a time. The default is to query the device for an optimal value.

**Specify `blocking_param` as:**

**`optimal`**

This is the default option.

The special non-numeric value `optimal` can be used to query the device for the value which it believes is optimal, which is the default. This is a good starting point, but if tape I/O is slow or there are other problems, you may want to increase this value for better throughput.

**`number`**

A number with memory units.

If only a number is specified, then the default unit of blocking is a “Block” and has a size of 512 bytes. This memory unit can also be specified in terms of bytes, kilobytes, megabytes, or gigabytes. Units can be abbreviated. For example, `-block 63k` specifies 63 kilobytes.

Depending on the device, this value may be subject to certain restrictions. For example quarter inch cartridge tapes often require the blocking to be a multiple of 512 bytes.

Whatever blocking was used to make a backup must also be used when restoring from that backup.

## (- comment) Backup comment

`-comment backup_comment`

Associate a comment with a backup. The default is no comment.

This option only makes sense when used with the `-backup` command and will be ignored when used with other commands. The comment specified will be stored with the backup and in the database and will appear when information about the backup is listed with the `-list` or `-info` commands.



---

A comment is useful for specifying the name or label of the tape on which the backup was made. When providing a comment, it is a good idea to enclose the comment string with single quotes. This prevents the shell from misinterpreting it as multiple arguments.

## **(- rollforward) Turn roll forward archiving ON**

`-rollforward`

When used in conjunction with the `-backup` command, turn roll forward archiving **ON** -but do not begin writing to the roll forward log.

After this option, you are guaranteed that no log records will be discarded unless they have been archived. Accordingly, if you do not later start roll forward logging with `-log`, you will sooner or later fill the log files to capacity. If a log file fills to capacity, the database will freeze until the log file is relieved by turning roll forward logging **ON** (which will cause the log entries to be archived and then discarded) or by terminating roll forward logging entirely (which will cause the log entries to be discarded after they have been applied to a database).

## **(- aggression) Aggression - Buffer time**

`-aggression aggression_param`

The maximum number of seconds' `vbackup` is allowed to hold data in a buffer before writing to tape.

The default interval is 60 seconds. Decreasing this number can make log records go to tape more quickly, but can decrease storage efficiency. Similarly, increasing the number increases the amount of time it takes log records to be written to tape and increases storage efficiency. This is because `vbackup` must write to tape only in chunks of the size specified by the `-blocking` parameter.

If the interval specified by `-aggression` elapses and there are still not enough unarchived log records to fill up one block, `vbackup` will add padding and flush anyway. Since this padding is wasted space, storage efficiency is reduced.

The `-aggression` flag can be used to specify either a single number for all databases being logged, or a list of numbers, one for each database.

## **(-script) Script**

`-script command`

If interaction is required, this option will perform the specified command and exit the current process with a message to the console.

This option is useful if you are running `vbackup` from a script. For example, if you are performing roll forward logging, this option can be used to notify you when an archive tape fills up.

If you are using UNIX, you can insert an entry into the `/etc/rc.local` file to start `vbackup` whenever the system reboots. This is best done by having `/etc/rc.local` run a script in a separate file which runs `setuid` to the DBA account. If you are using Windows, you can similarly run `vbackup` with an entry in a boot file. Following are relevant examples.

For example:

```
vbackup -script 'mail dba < vbackup.out' \
-log db1 db2 db3 > vbackup.out
```

The above command will start standard log archiving for three databases. If something happens which requires user input, such as the backup device becoming full, `vbackup` will exit and mail its output to the DBA user.

Another example:

```
i=0
while test $i -lt 30
do
    vbackup -dev log$i -script 'echo log full | mail dba' \
    -capacity 100M -log db1 db2 db3
    i='expr $i + 1'
done
```

The above command does the same standard log archiving. Rather than archiving directly to tape, it archives to disk in a series of 100 megabyte files named `log1`, `log2`, `log3`, etc. Whenever a new file is completed, the DBA is notified via mail. To avoid an infinite loop in the case when `vbackup` hits a legitimate error instead of a full tape, the script exits after 30 iterations.

There are many useful ways to use the `-script` option. For example, a line could be added which copied the 100M files to tape once it was completed. It could also be made to cycle through a series of tape devices rather than a series of files. Once one tape drive filled up, the next one would take over, thus minimizing the need for human intervention.

Following is an example of a script that allows you to switch between two disk files. When one file becomes full, it is closed, and a new file takes over while the previous file is being copied to tape.

---

```

    #Archive logs to file and copy the file to tape once it
    #reaches 100M. Delete the file once it is copied.
    #Limit ourselves to 30 iterations to avoid infinite loops.

blocking=10k
i=1
while [ $i -le 30 ]; do
    # backup logs to file
    vbackup -dev log$i -capacity 100M -blocking $blocking \
    -script 'echo log full | mail dba'-log group@gamehendge
    # wait until the last file was copied to tape
    if [ $i -gt 1 ]; then
        while [ -f log'expr $i - 1' ]; do
            sleep 5
        done
    fi
    # fork a process to copy the current file to tape
    (
        if dd if=log$i of=/dev/nrst0 bs=$blocking; then
            rm log$i
        else
            echo dd failed for log$i | mail dba
        fi
    ) &
    # increment i
    i='expr $i + 1'
done

```

## **(-getbeprofile) Extract Server Process Profile File**

-getbeprofile

This option is used in conjunction with the `-list` command to extract the backed up server process profile file `profile.be`.

`vbackup` backs up the server process profile file `profile.be` into the backup device or file. In case of multiple database backup, multiple server process profile files (`profile.be`) will get backed-up to the same backup device or file.

Use the `-getbeprofile` option to extract the server process profile file `profile.be` that was present at the time when the backup was taken. By default this option will dump the contents of all the backed-up server process profile files (`profile.be`) to stdout.

## **(-odir) Target Directory for Extracted Server Process Profile File**

```
-odir <output directory>
```

This option is used with the `-list` command and the `-getbeprofile` option together. The syntax is:

```
-list [-getbeprofile [-odir <path-to-save-profile>]]
```

Use the `-odir <output directory>` option to extract all the server process profile files (`profile.be`) into the specified directory. This directory should already exist and must have write permission. `vbackup` will create a new file under this directory that will adhere to the following naming convention –

```
<database name>_<hostname>_level<backuplevel>_profile.be
```

## **Changing the parameter settings**

If there is a recoverable error or an additional phase of restore (such as a second or third level of restore or a log roll forward), you can change certain settings after `vbackup` has been invoked.

The settings can be changed only for the following parameters: device, position, blocking, and capacity.

Following is an example, which shows how to change the device name after initially specifying a nonexistent device.

Suppose `vbackup` has been invoked as follows, and the device `/tmp/grp` does not exist:

```
#vbackup -dev /tmp/grp -backup group
```

In this case, the following output is seen (the version number will correspond to your particular release number):

```
VERSANT Utility VBACKUP Version 7.0.1.3
```

Copyright (c) 1989-2006 VERSANT Corporation  
Backing up database 'group' to device '/tmp/grp':

0%		50%		100%

Current settings are:

```
device    = '/tmp/grp'
position  = 'current'
capacity  = 'dynamic'
blocking  = 'optimal'
```

Could not open device '/tmp/grp' at position 'current'.

OS error #2. Please check the device. [?=help] ?

Change the setting by typing a name and new value (for example, "position 2"), type "quit" to quit, or press <Return> to proceed. It is only necessary to type enough of a command to make it unique, so you can type "q" instead of "quit" or "p 2" instead of "position 2".

After changing a setting you will see the following:

Change additional settings or type <return> to proceed. [?=help]

## Database Backup Frequency

A database should be backed up regularly. The more heavily a database is modified, the more frequently you will want to back up, either in full or incrementally. The basic trade-off is that full backups take longer than incremental backups, but they are much easier to use for a restore operation.

A rule of thumb for using backup levels is first to do a level 0 backup and then do level 1 backups. After a while, as changes accumulate, you will start to get tired of how long the level 1 backups are taking. At that point, you can start to do level 2 backups. When level 2 backups start to take a long time, then you should repeat the cycle by doing a level 0 backup.

If you do multiple level 1 and/or 2 backups, you should only use the latest one(s) when you restore the database.

For a heavily modified database, you might want to perform daily backups on a schedule similar to the following:

Day of the week:	Sun	Mon	Tue	Wed	Thu	Fri	Sat
Level of backup:	0	1	2	0	1	2	2

A less heavily modified database might be backed up daily, but with only one full backup:

Day of the week:	Sun	Mon	Tue	Wed	Thu	Fri	Sat
------------------	-----	-----	-----	-----	-----	-----	-----

Level of backup:    0       1       2       2       1       2       2

A lightly modified database could be backed up three times a week:

Day of the week:    Sun   Mon   Tue   Wed   Thu   Fri   Sat  
 Level of backup:    0       1       2

A very lightly modified database could be backed up weekly:

Week of the month:       1st   2nd   3rd   4th  
 Level of backup:           0       1       1       1

In short, you should back up at a frequency that is appropriate for the importance of a database and the frequency with which objects in it are modified.

If a database is small, you can perform a full level 0 backup every time. If it is large, you should take advantage of the incremental backup feature by setting the backup level to minimize the amount of time and storage associated with each backup.

If you want to restore the database to the most recently backed up state and if the database still exists, you can use the `-info` option to tell you the backup files and the order you need to use them to restore the database. If the database does not exist, you will have to figure out for yourself the order in which to use the backup files.

**See also “dbinfo” on page 150, and “stopdb” on page 220, which are utilities needed to perform restore operations.**

## Database Restoring Procedure

Following are the steps involved in restoring a database.

1. Determine what backup files you need.

Before restoring a database, you must determine which backup files you need. If you are recovering from a deletion or damaging of key objects, you can use the `-info` option to `vbackup` to list the set of backups necessary to restore the specified databases to their most recently backed up state. If list information is not available, you will have to rely on your own records.

**(See also the comments for the `-info` option.)**

2. Locate and examine files on tapes or other backup device.

---

Once you know what backups are needed, you must next locate the tapes or file devices on which they are stored. You can use the `-list` option to `vbackup` to list the backup files stored on a backup device.

**(See also the comments for the `-list` option.)**

3. Perform the restore.

To restore from a device, use the `-restore` option to `vbackup`. You must restore in the proper order as listed by `-info` or your own records, working from the last full backup to the most recent incremental backup.

**(See also the comments for the `-restore` option.)**

## vcopydb

```
vcopydb [options] dbname copy_dbname
```

Copy all objects and class definitions from one database to another.

After using this utility, the objects in the target database will have the same logical object identifiers as the objects in the source database.

While copying objects and class definitions from one database to another, please ensure that the same DBA password is specified for both the databases, in case the databases are associated with password-based authentication.

If the target database does not exist, use `makedb` to create the target database directory before running `vcopydb`. Both the source and the target databases must be group databases.

The copying of databases is a resource intensive operation and for databases sizes in the Gigabytes range, it is recommended to use special `profile.be` parameters of source and target database.

**Source database:** `max_page_buffs`, set value as high as possible.

**Target database:**

- The creation of data requires intensive logging, so tuning the log buffers and log files is essential for performance. So minimal recommended sizes are:
  - `llogvol 32M`
  - `plogvol 32M`

- llog\_buf\_size 128M
- plog\_buf\_size 128M
- Turn Locking off in the target database.
- The log\_buf\_size should be 4 \*logvolsize.

You can also alternatively use the `-optimize` option to tune the profile of the target database.

For more information, please refer to “**-optimize**”.

After changing the `profile.be` parameter stop the databases.

On completion of `vcopydb` please replace the `profile.be` of the database with the production settings and restart the databases.

**See also “createreplica” on page 145.**

**For platform specific requirements of `vcopydb` on AIX / Windows, please refer to the chapter “Platform Notes for AIX / Windows” in the *Release Notes*.**

### Parameters for this Utility are:

**dbname**

The name of the source database.

**copy\_dbname**

The name of the target database.

### Options for this Utility are:

**-batchsize M**

The default is 1000.

Copy M objects per transaction. In case there are large objects, there is a chance that user gets out of heap errors. You can use `-threads` and `-batchsize` option to keep the heap requirement low. If not specified `vcopydb` takes 1000 objects per transaction.

**-i**

Create, pre-allocate and initialize the target database.



---

A new database cannot be used as a replica database with the Fault Tolerant Server option.

#### **-nocreate**

Copy a database into an existing database without creating a new database for the copy.

An error will be returned if:

- The target database does not exist.
- The user does not have write access to the target database.
- The target database does not have enough space allocated to it to contain all objects in the origin database.
- The target database already contains user class definitions or objects. In this case, the error raised will be `UT_DB_NOT_EMPTY`.

**NOTE:-** `-nocreate` and `-i` cannot be used together.

#### **-nolock**

Turn Locking `off` in the source database.

#### **-noprint**

Suppress display messages.

#### **-noprogess**

Do not report the overall copy progress on this console. By default, the overall progress is displayed on the console via a progress gauge.

#### **-optimize**

Use automatic profile tuning for speedup.

Optimizes the copy database operation by automatically tuning the server profile (`profile.be`) of the target database for performance. Only the server profile of the target database is changed. The profile parameters are tuned before the start of the copy operation and restored after the copy operation completes.

Currently the following target server profile parameters are tuned automatically:

Parameter	Action
<code>locking</code>	This parameter is set to off

Parameter	Action
logging	This parameter is set to off

**NOTE:-** In case `vcopydb` terminates prematurely or is explicitly terminated, manual user intervention may be needed to restore the above mentioned server profile parameters to their original values. Additionally, the target database may not be in a usable state (since logging was turned off). Further steps would involve investigating why `vcopydb` terminated, fixing the problem and restarting `vcopydb`.

## **-threads N**

Copy a database using N threads. The number of threads is limited by system resources. If this parameter is not specified, `vcopydb` will optimize and create 3 to 30 threads by itself.

## vinfo

```
vinfo [ -l | -r | -v | -i ]
```

Display Versant product information, list the version information for each component or verify consistency of the installation.

### **Options for this Utility are:**

#### **-l**

Display Versant product patch number and component versions

#### **-r**

Display patch README. This works even if the patch Readme file has been removed.

#### **-v**

Verify checksum of all files inside binaries and headers.

Vinfo will report any files that do not have the expected checksum for the current patch level. Any files that are from older patch levels will be reported as such and any files that are completely unknown will be reported as corrupt.

For example,

---

```
% vinfo -v
ok odbms/demo/embeddability/ovbackup.c
ok bin/addvol
ok bin/cleanbe
error bin/cleanfe (This file matches the one shipped
                  with Versant patch 1)
ok bin/comparedb
ok bin/convertdb
ok bin/createdb
.....
error h/obmacros.h (This file matches the one shipped
                  with Versant base release, patch 1, patch 2, patch 3)
.....
```

#### **-i**

Identify the patch level of an installation by taking a checksum of each installed file and comparing with an internal database of checksums for all patch levels. After making the best effort identification, vinfo verifies that each file is correct for the assigned patch level.

**NOTE:-** vinfo only has knowledge of patch levels matching the current vinfo version and older.

For example,

```
% vinfo -i
This installation is Versant Object Database 7.0.1.3.
ok odbms/demo/embeddability/ovbackup.c
ok bin/addvol
ok bin/cleanbe
error bin/cleanfe (This file matches the one shipped
                  with Versant patch 1)
ok bin/comparedb
ok bin/convertdb
ok bin/createdb
.....
error h/obmacros.h (This file matches the one shipped
                  with Versant base release, patch 1, patch 2, patch 3)
.....
```

## verrr

```
verrr err_num|err_name|err_frag
```

Print an error message given an error number, name or fragment.

### Parameters for this utility are:

#### **err\_num**

Specify the error number of error message to print.

#### **err\_name**

Specify the error name of the error message to print.

#### **err\_frag**

Specify a fragment of the error message to print.

The `verrr` command reads error messages from the file `error.txt` in the `VERSANT lib` directory.

The `error.txt` file also contains explanations of many errors.

Programs linked with Versant, read error messages from the file `error.msg` in the `VERSANT lib` directory.

The `error.msg` file contains only one-line error messages.

Following are several examples of using the `verrr` utility to look up an error using a number or name. The entry in bold is the command statement and following are the lines returned.

```
verrr 2903
===== SEARCHING FOR '2903' =====
2903, SM_LOCK_TIMEDOUT: Lock wait Timed out

verrr SCAP_PREP_CLS
===== SEARCHING FOR 'SCAP_PREP_CLS' =====
8127, SCAP_PREP_CLS: Cannot add signature to class
      object,for class %s
We were unable to mark the class object as dirty,
so we could not add the class signature to it.
This will prevent the schema signature assertion at
runtime.
```

Note that for the second example, explanatory information was available.

If you abbreviate the `err_name` argument, all messages matching the partial name will be printed.

Example:

```

verr OUT_OF
===== SEARCHING FOR 'OUT_OF' =====
1022, SM_E_OUT_OF_CLASS_CB_SPACE: Out of class CB
                                   space
1033, SM_E_OUT_OF_CSR_SPACE: Out of cursor space
1083, SM_E_OUT_OF_VOL_SPACE: all volumes exhausted
8816, EVPP_OUT_OF_MEMORY: out of memory in parser
The vpp program ran out of memory. Adding more swap
space could help, as could removing some processes
from the machine.
Using "stopdb" on a Versant database can also help a
lot.
```

## verrindx

verrindx

Any time you define your own error message and alter the file `error.msg`, you must re-index it.

Do this by running the Versant `verrindx` utility. This utility takes no command line arguments and prints nothing while running.

## vinstinf

vinstinf

This utility will print the details of the installation information needed for generation of the license keys.

Machine type and machine identification (`Hostid`, `ipv4` or `ipv6`) data is needed when requesting for a license key. Any arguments passed will be ignored and will print the following details on the machine it is executed.

1. Machine name

2. Machine type
  1. For Sun machine: `SunOS`
  2. For Windows machine: `Windows`
  3. For IBM machine: `AIX`
  4. For HP machine: `HP-UX`
  5. For Linux machine: `Linux`
3. Hostid
4. IPv4
5. IPv6
6. Total number of CPU cores

## vlicchk

```
vlicchk -c <component_name> -v <component_version> [ <server> ]
```

The function of this utility is to determine whether a specified component is licensed.

### Parameters for this Utility are:

**-c <component\_name>**

The component name as decided by the component team.

**-v <component\_version>**

The most significant two digits identifying the version of the component. This should follow the component specification. e.g. 6.0

### **server**

The server machine to contact for performing the license check. If this parameter is not specified, the utility will perform a local license check. If a server name/ address is specified is the same machine on which the utility is executed, the utility will perform a local license check, else it will perform the remote license check.

## vlicvrfy

```
vlicvrfy [ <license-file> ]
```

The function of this utility is to scan a specified license file for errors.

It will accept one optional argument, which will be the license file to be scanned. In case no arguments are specified, the following table identifies the files to be scanned:

Order	Environment Variable	License File Name
1	VERSANT_LICENSE_FILE	\$VERSANT_LICENSE_FILE
2	VERSANT_ROOT	\$VERSANT_ROOT/ license.xml
3	ENJIN_ROOT	\$ENJIN_ROOT/ license.xml
4	TPEROOT	\$TPEROOT/license.xml

## vmovedb

```
vmovedb [options] [-C <c1> <c2> <c3> ...] <src_db> <target_db>
```

Move all objects and copy class definitions schema, from source database to target database.

### Parameters for this Utility are:

#### src\_db

The name of the source database from which objects/schema are to be moved.

#### target\_db

The name of the target database into which objects/schema are to be moved.

### Options for this Utility are:

```
-C <c1> <c2> <c3> ...
```

This is the class name parameter. If not specified, all the objects from non-system classes will be moved to the target database else `c1`, `c2`, `c3` are the classes whose objects are to be moved.

If a class with zero instances has been specified by the user, vmovedb will copy only the schema.

If the source database is empty, vmovedb will return the err UT\_NO\_OBJS\_TO\_MOVE.

## **-subclasses**

This will be specified on the command line if the user needs to move objects from all the subclasses for the selected class(es). If not specified, only objects belonging to the class and not its subclasses will be moved.

## **-closure L**

For the selected set of objects, move, objects in the closure set taken up to L levels. If this option is not specified the value of L is '0' i.e. no objects linked to that object (either directly or indirectly) will be moved. Specify -1 to return all objects linked directly or indirectly to the objects selected by the -C option.

## **-threads N**

Move objects in a database using N threads. The number of threads is limited by system resources. Increasing `plogvol` and `max_page_buffs` in `profile.be` and turning off locking (for the source database) will also improve scalability. If N is not specified, vmovedb will optimize and create 3 to 30 threads by itself.

## **-batchsize S**

The default is 1000.

Move S objects per transaction. If not specified, then vmovedb will move 1000 objects per batch. This feature will improve scalability of vmovedb.

## **Usage Notes:**

vmovedb is not allowed on a replica pair of databases. An attempt to do so will result in error UT\_NO\_REPLICA\_MOVE.

vmovedb requires that the target database be in existence. If it does not exist, error TARGET\_DB\_NOT\_CREATED will be returned.

If the source and target databases are the same, vmovedb returns error UT\_VMOVEDB\_INVALID\_TARGET.

If the source database is associated with a password, then the same password should be specified for the target database. If the password verification fails, SM\_E\_INVALID\_PASSWORD error is returned.



---

## vstats

```
vstats [ options ] [ command ]
```

Start the Statistics Tool.

The `vstats` utility can collect and view statistics that will tell you where time is being spent by your application process and the server process for each connected database.

**For more information, on statistics collection and usage notes, please refer to the chapter “Statistics Collection” in the *Versant Database Fundamentals Manual*.**

## Commands

**Command Parameters for the Utility are:**

### List statistics

#### **-summary**

List all defined statistics along with a short explanation of each one.

If this alternative is given, all other parameters are ignored.

#### **-stats *expr\_list***

Specify a list of statistic expressions for viewing. The items in the list are separated by spaces.

The specified statistics in *expr\_list* will be read from a file or from a direct connection per the `-filename`, `-stdin` and `-database` parameters.

Statistic names must be specified as shown in the section "Function and Statistic Names", except without the `STAT_` prefix and transposed to lower case. Derived statistics cannot be used directly, instead specify the basic statistic names. To see the available names online, use the `-summary` command.

For connection and database statistics (those with the prefixes `be_` and `db_`), you must also use a database name since the same statistics may be available for multiple databases. The general syntax is:

```
"statistic_name database_name"
```

For example:

```
vstats -database MasterDB -on db_obj_received db_obj_sent
vstats -database MasterDB -iterations 1000 -stats "db_obj_received M
asterDB" "db_obj_sent MasterDB "
vstats -database MasterDB -off db_obj_received db_obj_sent
```

You can use simple mathematical formulas to combine and modify statistics to produce a derived statistic.

For example:

```
vstats -database MasterDB -stats "db_obj_received MasterDB +
db_obj_sent MasterDB"
```

In these formulas you can use the standard numerical operators +, -, \*, /, (, and ). You can also use the following special functions:

delta — The difference between an expression's current value and its next most recent value.

min — The minimum value of an expression over the `vstats` run.

max — The maximum value of an expression over the `vstats` run.

avg — The average value of an expression over the `vstats` run.

For example to view traffic to and from a database named groupdb on the connection id 20 in objects per second, the following command could be used:

```
vstats -database MasterDB -id 20 %1 -stats "delta (
be_obj_sent MasterDB + be_obj_received MasterDB ) / delta be
_real_time MasterDB"
```

## **-timestamp**

The `vstats` command `-timestamp` has been deprecated from VOD 7.0.1.4. The timestamp will always be displayed when viewing the statistics.

To print timestamp along with the statistics. This is the default option for each output of `vstat`.

The default format for the timestamp is HH:MM:SS.

The timestamp format can be changed by passing new format as parameter to `-timestamp`.

The format argument can consist of one or more codes. The formatting codes are preceded by a percent sign (%). Characters that do not begin with % are printed unchanged.

The format argument varies for different operating systems as it is implemented using OS specific function call. For details of format argument please refer to the documentation of strftime C function on the specific operating system.

Following are few format arguments that are common for most of the operating systems:

<b>Format specifier</b>	<b>Interpretation</b>
%a	Abbreviated weekday name as per current locale
%A	Full weekday name as per current locale
%b	Abbreviated month name as per current locale
%B	Full month name as per current locale
%c	Date and time representation appropriate for locale
%d	Day of month as decimal number [01 – 31]
%H	Hour in 24-hour format [00 – 23]
%I	Hour in 12-hour format [01 – 12]
%j	Day of year as decimal number [001 – 366]
%m	Month as decimal number [01 – 12]
%M	Minute as decimal number [00 – 59]
%p	Current locale's A.M./P.M. indicator for 12-hour clock
%S	Second as decimal number [00 – 59]
%U	Week of year as decimal number, with Sunday as first day of week [00 – 53]
%w	Weekday as decimal number [0 – 6; Sunday is 0]
%W	Week of year as decimal number, with Monday as first day of week [00 – 53]
%x	Date representation for current locale
%X	Time representation for current locale
%y	Year without century, as decimal number [00 – 99]
%Y	Year with century, as decimal number
%Z	Either the time-zone name or time zone abbreviation, no characters if time zone is unknown

%%                      Percent sign

### Examples

- To print the default timestamp along with db\_cache\_hit\_ratio statistics, vstats can be invoked as follows:

```
vstats -timestamp -database MasterDB -stats 'db_cache_hit_ratio
MasterDB'
```

- To print the timestamp in 12 hour format, vstats can be invoked as follows:

```
vstats -timestamp "%I:%M:%S %p" -database MasterDB -stats 'db_cache
_hit_ratio MasterDB'
```

%I      signifies print hour in 12-hour format [01 – 12]  
 %M      signifies print minute as decimal number [00 – 59]  
 %S      signifies print Second as decimal number [00 – 59]  
 %p      signifies print current locale's A.M./P.M. indicator for 12-hour clock

As ':' are not preceded by '%' sign, they will be printed unchanged.

Thus output of above command would be similar to following:

```
VERSANT Utility VSTATS Version 7.0.1.3
Copyright (c) 1989-2006 VERSANT Corporation
```

```
T = Timestamp
0 = db_cache_hit_ratio MasterDB
```

```

      T           0
=====
11:31:03 PM      -
```

```
11:31:08 PM      -  
11:31:13 PM      -
```

## Profiling on

```
-on [ stats_list ]
```

Turn automatic profiling on for connection and database statistics.

If no statistic names are specified, all connection and database statistics will be turned on.

If a list of statistic names is specified in `stats_list`, collection will be turned on only for the specified statistics. The items in the list are separated by space. Normally, you will want to specify only connection statistics (with a `be_` prefix) and database statistics (with a `db_` prefix).

Statistic names must be specified as shown in the section "Function and Statistic Names", except without the `STAT_` prefix and transposed to lower case.

To see the available names online, use the `-summary` command.

When using `-on`, you must specify a database with the `-database` option.

This means that in a list of statistic names, you do not use "`stat_name dbname`" syntax as in the `-stats` command.

When using `-on`, you may also specify a connection with the `-id` option. If a connection is not specified, connection statistics will be turned on for all connections to the database at the time `vstats` is invoked. If additional connections are made at a later time, you must invoke `vstats` with `-on` again to turn automatic profiling on for the new connections.

The on state is not persistent. If a database is stopped and restarted, automatic profiling reverts to the default state of off.

Some statistics, particularly those that measure time intervals, are expensive to collect. Accordingly, you should turn on as few statistics as possible when performance is an issue. Similarly, when collection of a statistic is no longer necessary, you should turn collection off. Turning a statistic off resets its value to 0.

## Profiling off

```
-off [ stats_list ]
```

Turn automatic profiling off.

This command can be used only by the DBA user who created the databases being monitored.

If no statistic names are specified, all statistics will be turned off. If a list of statistic names is specified in `stats_list`, collection will be turned off only for the specified statistics. The items in the list are separated by space.

Statistic names must be specified as shown in the section "Function and Statistic Names", except without the `STAT_` prefix and transposed to lower case. To see the available names online, use the `-summary` command.

When using `-off`, you must specify a collection database with the `-database` option.

You may also specify a connection with the `-id` option. If a connection is not specified, connection statistics will be turned off for all connections to the database at the time `vstats` is invoked. If additional connections are made at a later time, you must invoke `vstats` with `-off` again to turn automatic profiling off for the new connections.

If a database is stopped and restarted, automatic profiling reverts to the default state of off.

## Get connections

`-connections`

For each connection associated with the database specified in the `-database` option, print the connection identifier, username, session name, server process identifier, application process identifier, and protocol information.

This command is useful to get a connection identifier that can be used with the `-id` option.

## Get locked objects

`-locks`

For each lock held in the database specified in the `-database` option, print the logical object identifier for the locked object, the external lock mode, the internal lock mode, the transaction identifier and whether the object has a pending lock request.

## Get transactions

`-transactions`

For each transaction with a connection to the database specified in the `-database` option, print the transaction identifier, number of locks held by the transaction, the connection identifier associated with the transaction.

## Get list of enabled statistics

`-list`

This command gives the list of enabled (ON) statistics on the server side, i.e backend statistics and database statistics, for a particular database.

Example:

- To print the list of enabled statistics, `vstats` can be invoked as follows:

```
vstats -d dbname -list
vstats -d dbname -id <connection-id> -list
```

**NOTE:** - For the backend statistics, it is mandatory to specify the connection id.

## Usage

- The scope of `vstats` is all database connections in existence at the time it is invoked. If a database connection is made after `vstats` is invoked, statistics collection for that connection will be on or off per the default set in the stat database configuration parameter.
- The `vstats` utility will send statistics to `stdout`. To send statistics to a file, you must collect statistics with an application and set the `VERSANT_STAT_FILE` environment variable to point to a file name.

## Sample Output

Following is sample output for several runs of `vstats` with various command options:

```
vstats -d group -connections
Connection ID to database "group":
Connection ID      = 417
User Name          = "george"
Session Name       = "vstats -connections"
Long Transaction   = "0.0"
Server Process     = "gamehendge":417
Client Process     = "gamehendge":417
Protocol           = TCP/IP
Server Port        = 192.70.173.25:5019
Client Port        = 192.70.173.25:4049

vstats -d group -transactions
```

Transactions in database "group":  
Transaction ID = "10432.0.3076"  
Name = "vstats -transactions"  
Lock Count = 2  
Connection ID = 418  
Flags = "Short"

vstats -d group -locks  
Locks in database "group":  
Object = "10432.0.1027"  
External Mode = "Intention Write"  
Internal Mode = "Intention Write"  
Transaction ID = "10432.0.4100"  
Flags = "Running, Transient"  
Object = "10432.0.2052"  
External Mode = "Read"  
Internal Mode = "Read"  
Transaction ID = "10432.0.4100"  
Flags = "Running, Transient"  
Timestamp output:

VERSANT Utility VSTATS Version 7.0.1.3  
Copyright (c) 1989-2006 VERSANT Corporation

T = Timestamp  
0 = delta db\_data\_located sek1  
1 = delta db\_data\_reads sek1  
2 = delta db\_data\_writes sek1  
3 = delta db\_obj\_sent sek1  
4 = delta db\_obj\_received sek1  
5 = delta db\_net\_rpcs sek1  
6 = delta db\_qry\_btree\_objs sek1  
7 = delta db\_qry\_scan\_objs sek1  
8 = delta db\_checkpoints sek1  
9 = delta db\_xact\_committed sek1

T	0	1	2	3	4	5	6	7	8	9
===	===	===	===	===	===	===	===	===	===	===
18:15:58	-	-	-	-	-	-	-	-	-	-
18:15:59	0	0	0	0	0	1	0	0	0	0



---

18:16:00	0	0	0	0	0	1	0	0	0	0
18:16:01	0	0	0	0	0	1	0	0	0	0
18:16:02	0	0	0	0	0	1	0	0	0	0
18:16:03	0	0	0	0	0	1	0	0	0	0
18:16:04	0	0	0	0	0	1	0	0	0	0
18:16:05	0	0	0	0	0	1	0	0	0	0

## Viewing Input Options

When you want to view statistics with the `-vstats` command, you can view them either from a file or from `stdin`. Following are options for the source of the statistics to be viewed. If none of these options are specified, the file specified in the `VERSANT_STAT_FILE` environment variable will be used. If `VERSANT_STAT_FILE` is not set, `stdin` will be used. The viewing options are mutually exclusive.

Viewing options are:

### Source file

`-filename filename`

Read from the specified statistics file.

### Source stdin

`-stdin`

Read from `stdin`.

The `-stdin` option can be used to view a profile file as it is being generated. For example:

```
tail +0f file | vstats -stdin -stats se_reads
```

If you use `-stdin` to read a file, you will probably want to use the `VERSANT_STAT_FLUSH` environment variable.

### Source database

`-database dbname`

Read directly from the specified database with a direct, real-time connection.

### Username

`-username user`

User name to access the DB. This user has to be a valid user of the DB . You will be prompted for a password if no password is entered using the `-password` option. Username cannot be more than 32 characters in length.

The default is to run as the OS user without any password.

## Password

`-password passwd`

Password corresponding to the user name entered in `-username`. This is mainly intended for scripts so that users don't have to enter the password interactively. For normal usage, users can just pass in the `-username` option and `vstats` will prompt them for the password. Passwords cannot be more than 256 characters in length.

Running with just the `-password` option without the `-username` option will return an error.

## Source connection

`-id connection-id`

Read only statistics related to the specified connection to the database specified in the `-database` option.

By default the connection associated with the `vstats` process is used. For this reason, you should use the `-id` option if you are using `-database` and have specified any connection statistics (otherwise, you will just get the statistics for the `vstats` connection).

You can use the `-connections` command with the `-database` option to get connection identifiers.

You can use only one of the three options `-filename`, `-stdin`, and `-database`.

To view statistics with a direct connection using the `-database` option, you must first enable those statistics, which you are interested in viewing using the `-on` command. You can then view them with the `-stats` command.

## Viewing Output Options

When you want to view statistics with the `-stats` command, you can pick one or more of the following options to specify the information to be displayed.

---

## No information

`-noinfo`

Do not show the information column. By default, the last column in the output is used to show the function name or comment associated with the current set of statistics.

## Pause

`-period seconds`

Pause operation for the specified number of seconds between each line of statistics. When reading from a file specified with `-filename` or from stdin specified with `-stdin`, the default pause is 0 seconds. When reading directly from a database specified with `-database`, the default pause is 5 seconds.

## Iterations

`-iterations iter`

Display the specified number of statistics readings. The default of 0 means to output all readings in the file or, in the case of direct reading from a database, to take readings indefinitely.

## Statistics File

When it starts, `vstats` will read a configuration file in which derived statistics can be pre-defined and named. On UNIX installations, the name of the file is `.vstatsrc`. On personal computer installations, the name of the file is `vstatsrc`.

A system-wide `vstatsrc` file in the `VERSANT lib` directory will always be read. If a file named `.vstatsrc` exists in your UNIX home directory, it will also be read. These files can contain comments beginning with `#` and commands to define new derived statistics. See the "Directories and Files" chapter for information about `.vstatsrc` and `vstatsrc`.

## Fault Tolerant Server option

Even when synchronous database replication has been turned on, statistics collection tools and mechanisms operate only on named databases and not on replica databases. To collect statistics for a replica database, you must apply statistics collection mechanisms specifically to the replica database. If the replica database goes down, statistics collection will stop and then restart when the database comes back up.

**For more information, please refer to the *Versant Veding Usage Guide*.**

## Sample Output

When you use the `-stats` command to read a file or view a direct connection, you will see a header and a listing.

In the header, the names of the statistics being analyzed will be printed, along with explanations of each statistic.

In the listing, the statistics values themselves will be presented in columns. By default, there will be an additional column indicating a function name or comment associated with the statistics. If the statistics are coming from a file, some additional information may also be provided. If a statistic is undefined, a dash will be printed in place of a value until that statistic becomes defined.

The following is example of output when the `-stats` command is used.

Note the misspelling of the entry "se\_dirtyy."

```
vstats -stats se_objs "delta se_objs" se_dirtyy "db_net_reads group2"
```

```
VERSANT Utility VSTATS Version 7.0.1.3
```

```
Copyright (c) 1989-2006 VERSANT Corporation
```

```

0 = se_objs
1 = delta se_objs
2 = se_dirtyy
3 = db_net_reads group2
0          1          2          3
=====
109        -          -          -
109         0          -          -      o_xact(group)
140        31          -          -
58        -92          -          -      o_xact(group)
219       161          -          -
58       -161          -          -      o_xact(group)
```

In the above example, Column 0 shows values for the `se_objs` statistic and Column 1 uses the `delta` function to show the difference in this value from reading to reading. Note that the first entry in Column 1 is a dash, since `delta se_objs` is undefined until two readings are available.

Columns 2 and 3 are undefined all the way down. In the case of Column 2, this is because the statistic `se_dirtyy` is undefined due to a misspelling.

In the case of Column 3, entries are undefined, because no statistics were ever collected for a database called `group2`. There are many possible reasons for this. For example, perhaps no

connection was ever made to `group2`, or the `be_net_reads` statistic was never turned on, or the `VERSANT_STAT_DBS` environment variable was used to disable collection of statistics for `group2`.

The last column shows some additional information recorded by the automatic collection mechanism. Each two lines of output show statistics recorded at the very beginning and very end of a function call to the database server process. On the second line of each of these pairs, the name of the function for which the statistics were recorded is shown, as well as a list of databases from which backend and database statistics were collected. This information column is not available when the direct connection model is being used or when a time interval limiting automatic collection is specified, for example, by using the `VERSANT_STAT_TIME` environment variable. The display of the information column can be suppressed by using the `-noinfo` option.

The following sample output is for a case where detailed statistics were collected for a function named `amalgamateObjs()`, which selects objects from two databases, reads the objects into memory, and performs a commit. For this case, assume that `amalgamateObjs()` has been defined to invoke the following functions:

No.	Function Invoked	Output Occurs
1.	<code>o_autostatsenter()</code>	on entry into <code>amalgamateObjs()</code>
2.	<code>o_select()</code>	on entry and exit
3.	<code>o_greadobjs()</code>	on entry and exit
4.	<code>o_autostatswrite()</code>	on invocation: a comment is printed
5.	<code>o_select()</code>	on entry and exit
6.	<code>o_greadobjs()</code>	on entry and exit
7.	<code>o_autostatswrite()</code>	on invocation: a comment is printed
8.	<code>o_autostatswrite()</code>	on invocation: a comment is printed
9.	<code>o_xact()</code>	on entry and exit
10.	<code>o_autostatswrite()</code>	on invocation: a comment is printed
11.	<code>o_autostatsexit()</code>	on exit from <code>amalgamateObjs()</code>

In this case, `vstats` was invoked with the following:

```
% vstats -stats fe_real_time "db_net_reads db1"
      "db_net_reads db2"
```

After the invocation of `vstats`, the application was run and called `amalgamateObjs()`, which caused the following output. Column 1 shows elapsed time. Columns 2 and 3 show the number of object read. The ">" symbols in the `info` column show the nesting of function calls. The annotation

comments in italics would not appear in the actual output and have been added here for explanation.

VERSANT Utility VSTATS Version 7.0.1.3  
 Copyright (c) 1989-2006 VERSANT Corporation  
 Statistics collected: Thu Jul 14 22:30:50 2005

fe\_real\_time = Seconds elapsed  
 db\_net\_reads = Reads from frontend  
 Column 1 = fe\_real\_time  
 Column 2 = db\_net\_reads db1  
 Column 3 = db\_net\_reads db2

1	2	3	info	
0.451	0	0		<- o_autostatsenter()
0.603	0	0	>	<- o_select() entry
1.452	1	0	>o_select(db1)	<- o_select() exit
1.612	1	0	>	<- o_greadobjs() entry
3.123	3	0	>o_greadobjs(db1)	<- o_greadobjs() exit
3.523	3	0	>got objs from db1	<- o_autostatswrite()
3.602	3	0	>	<- o_select() entry
4.124	3	1	>o_select(db2)	<- o_select() exit
4.298	3	1	>	<- o_greadobjs() entry
5.011	3	3	>o_greadobjs(db2)	<- o_greadobjs() exit
5.128	3	3	>got objs from db2	<- o_autostatswrite()
7.128	3	3	>done amalgamating	<- o_autostatswrite()
7.278	3	3	>	<- o_xact() entry
9.876	1	12	>o_xact(db1 db2)	<- o_xact() exit
	1			
9.978	1	12	>committed changes	<- o_autostatswrite()
	1			

```
10.011 0 0 amalgamateObjs(db1 db2) <- o_autostatsexit()
```

## vstream

```
vstream mode -d dbname options filename
```

Export the contents of a database to a file or import data from a file into a database.

### Parameters for this Utility are:

#### **mode**

Import or export the database.

#### **-d dbname**

Name of database to receive or supply data.

#### **options**

Import or export options.

#### **filename**

Name of file to receive or supply data.

## Mode

Alternatives for the mode parameter are:

Mode	Description
-i	Stream data from a file to a database.
-o	Stream data from a database to a file.

### **Input mode, -i**

In input mode, vstream reads in objects according to the exported data file and creates a new logical object identifier (loid) for each object. The `-p` option allows you to use the same loid's as in the original database.

## Output mode, -o

In output mode, vstream iterates through all the objects in the database using cursors and streams the objects to a file. You can use the option `-n` to set how many objects to read in each cursor fetch. The default setting limits the number of objects read in one cursor fetch to not exceed 80% of the cache size.

## Options

### Options for this Utility are:

#### `-p`

Preserve object links when importing.

The `-p` option ensures that logical object identifiers are not changed during the import process.

When importing data from a file to a database in multiple executions of vstream, you must use the `-p` option to preserve link relationships.

In general, you should never break an object graph into two vstream calls, or you may not be able to properly restore it.

For example, suppose you have a container containing classes A and B and that instances of these classes have links to one another. Now, suppose that you use two vstream iterations to export the database:

```
vstream -o -d db -c A data1.dat  
vstream -o -d db -c B data2.dat
```

Then, later, if you import these two database files into a database with the following commands, the link relationships between A and B will not be set properly:

```
vstream -i -d db data1.dat  
vstream -i -d db data2.dat
```

However, link relationships will be set properly with the following commands, because the logical object identifiers were preserved:

```
vstream -i -p -d db data1.dat  
vstream -i -p -d db data2.dat
```



---

When exporting data, if logical object identifiers are to be preserved in a destination database, you should either filter by class name or set level filtering to 1. If level filtering is set to a number greater than 1, some duplication in the data stream might occur.

**-a**

Aborts on schema miss-match. This option is applicable when input mode ( -i ) is enabled.

With this option, `vstream` will abort if the schema from the stream differs for the existing classes in target database.

**-t num**

Number of import or export iterations to perform before executing a commit and releasing the object cache.

The default is one iteration, which means `vstream` releases memory after each cursor fetch or after reading each `vstr` from an exported data file.

**-n num**

Number of cursors to be used to fetch objects.

**-l level**

Maximum levels of an object graph to export.

The default is -1, which means read the entire object graph.

Specify a positive number to specify a particular number of levels to export. Setting a low level number, such as 1, will reduce memory requirements.

If you specify the option -1 (read entire graph), `vstream` will write the morphology data to a file. The name of this file will be the same as your data file, except with the suffix ".morph". This file must be present for `vstream` to recreate the database.

**-g**

Turn on the group read option.

This option will improve performance if your object graph (the object tree) is not very complex.

**-f**

Fast mode, which means read all objects into memory when exporting.

This option improves performance but requires a lot of memory.

## **-nolock**

Set default lock to `NOLOCK` during export

## **-c class**

Specify names of the comma separated classes whose instances will be exported. While specifying the multiple class names, there should not be any space after the comma separator.

In case of template classes with multiple parameters, the comma separating the template parameters should be escaped with `\` escape character. Additionally, if the list of classes contains a template class then it should be specified using double quotes.

For example, to stream out a class A and a template class B having two template parameters you need to use the following command line:

```
vstream -o -d db -c "A,B<o_4b\,o_4b>" data1.dat
```

To stream out template class B you can use;

```
vstream -o -d db -c "B<o_4b\,o_4b>" data2.dat
```

To stream out class X, class Y and class Z you can use;

```
vstream -o -d db -c X,Y,Z data3.dat
```

If you specify the `-c` option, `vstream` will write morphology data to a file. The name of this file will be the same as your data file, except with the suffix `".morph"`. This file must be present for `vstream` to recreate the database.

## **-q query**

Specify instances to be exported by using a query.

The query should have the general form `"class attribute op value"`

where:

`class` is the class name,

`attribute` is the attribute name,

`op` is one of `{=, <, >, <=, >=, !=}`, and

`value` is the attribute value, which must be matched.

Only objects matching the query will be exported.

---

Place the query in quotes so that it is parsed as a single argument.

**-q7 vql7\_query**

Specify instances to be exported by using a VQL7 query.

Projections are not supported.

Only objects matching the query will be exported.

Place the query in quotes so that it is parsed as a single argument.

For example, to stream out all “Hello” objects that are pointing to a particular instance, you can use the following command line:

```
vstream -o -d db -q7 "select selfoid from Hello where Hello::mylink  
=7.0.3086" data4.dat
```

**For more information on VQL7 queries, please refer to the chapter “Query Processing with VQL 7.0” in the *Versant Database Fundamentals Manual*.**

**-u user**

Run as the specified DB user.

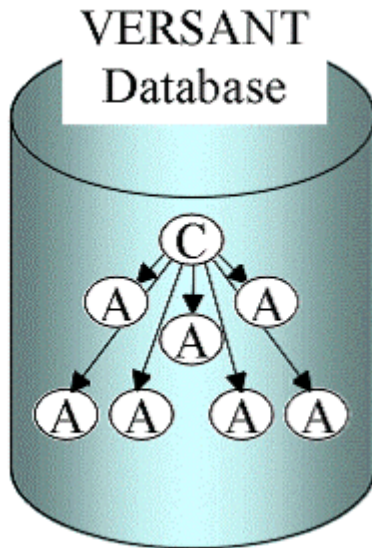
**-pwd password**

Enter password for the DB user specified in -u.

## Specifying levels to export

When an object is streamed out, by default the whole object graph referenced by this object is streamed. If the object graph is very large, as is the case for a container object that has links to every object in the database, level filtering can be used to control the depth of the object graph.

For example, streaming out a database with level 1 will export all the top level objects without following any further links; streaming with level 2 will force `vstream` to export an object and any object that is linked to it. The internal morphology map will guarantee the consistency of the object graphs.



```
vstream -o -l 1 = export <Object C>
vstream -o -l 2 = export <Object C> + 7*<Object A>
```

Within one `vstream -l` execution, all the object relationships (morphology) are preserved.

## Synchronizing schema

When you are importing into a database, if a class schema for the imported classes already exists in the target database, `vstream` will try to synchronize the class definitions in the new database.

The attributes in the stream schema and the database schema must be equal in name, type, and repetition factor for a successful match. The following schema synchronization rules apply:

- If the schema defined in the new database is the same as in the stream data file, no synchronization is done
- Any object attributes present in the database schema and not present in the stream schema will not be overwritten
- Any object attributes present in both the database and stream schema is overwritten by the stream value. Attributes do not have to be in the same sequence
- Any linked attributes present in the stream but not in the database are still processed, but not reproduced. The sub-graph of the link is still read in

- Any non-linked attributes present in the stream but not in the destination database are ignored

## Operating across platforms and releases

You can stream data with `vstream` across platforms. For example, you can stream out a HP database and then restore it on a Solaris machine.

Data can be streamed across different versions of Versant. However, when you stream from a newer versioned database to a older versioned database, make sure that you don't have any version-related system class inside your new database that cannot be handled in the old database.

## Estimating operating system resources

- In general, close as many applications as possible before running `vstream`.

The `vstream` utility requires adequate random-access memory (RAM) and a swap file of sufficient size to hold the morphology data. You can estimate the operating system (OS) resources needed for the streaming process for a given database size.

- If the page fault number remains high for over five minutes, `vstream` probably does not have enough RAM.

Windows users :

For the current page fault number, refer to `Task Manager>Processes>Page Faults`; if it changes quickly, little progress is probably being made.

To export when memory is low, you may need to export only top-level objects by using the `-l 1` option.

### Example —

Following is an example that calculates resources based on Windows operating system requirements. The same equations are used to generate the resource requirements for a UNIX system.

The first step in estimating resources is to consider the object graph. The complexity of a given database object graph depends on the number of links between objects. In a worst case graph, each object is linked to every other object; in a best case graph, each object is isolated.

The table below lists object graph scenarios and the typical amount of RAM needed according to the number of objects,  $x$ , in the graph.

### Graph Complexity —

Scenario	RAM	Swap File
----------	-----	-----------

---

Worst Case	40x	40x
Average Case	20x	40x
Best Case	4x	40x

The swap file must be large enough to contain the entire morphology map, which is always 40 times the number of database objects.

In the worst-case scenario, an average object has links to many other objects. This requires references to object information throughout the morphology map and during the streaming process. To achieve an acceptable level of performance, the required RAM size is approximately 40 times the number of objects plus the amount of RAM needed by the operating system.

In the average and best case a scenario, the object graph is relatively sparse and objects once processed are not referenced again. You still need the same swap file size, but the amount of RAM required is reduced to half or less. You can stream databases with smaller amounts of RAM if a typical object is referenced only a few times. Once these objects are swapped to disk by the operating system, it is less likely that they will be needed again. In the worst-case scenario, however, many objects are frequently referenced and the insufficient available RAM can cause page faults.

The following formulas can be used to estimate the amount of operating system resources (RAM and swap file) needed:

Required RAM = (factor \* ObjectCount) + OsRAM  
Required swap file size = (40 \* ObjectCount) + OsSwap

Elements of the above equations are:

Elements	Description
factor	Based on the morphology scenario (see the Graph Complexity table above).
ObjectCount	The approximate number of database objects. You can calculate number of database object by running the db2tty utility.
OsRAM	The operating system requirement for RAM.
OsSwap	The operating system requirement for the swap file.

The following examples illustrate this method of estimating operating system resources in two cases. The operating system sizes in the examples refer to Windows. The RAM and swap file requirements are similar on other supported platforms.

#### Example Operating System Resource Requirements—

Database size	400 MB	1 GB (4)
Object count (1)	2 million	10 million
Case	worst	Average to Best
Factor	40	15 (5)
RAM (2)	$(40 \times 2 \times 106) + 30\text{MB} = 110\text{MB}$	$(15 \times 107) + 30\text{ MB} = 180\text{MB}$
Swap file (3)	$(40 \times 2 \times 106) + 100\text{MB} = 180\text{MB}$	$(40 \times 107) + 100\text{MB} = 500\text{MB}$

#### Tips for calculating resources:

1. Can be calculated using `db2tty`.
2. Most databases will not require this maximum RAM amount. A test using a highly connected or worst case database of 380MB ran perfectly with a 96MB RAM limit. The Windows OS alone requires 30MB RAM. If additional applications are running, increase this value. Vary this value for other operating systems accordingly.
3. The Windows OS alone requires 100MB swap file. Additional running applications will require an increased swap file size. Vary this value for other operating systems accordingly.
4. With a database of this size, it is likely that the graph is sparse, so the user should consider this an average or best case scenario.
5. Since this database is large and is between the suggested values for average (20) and best (4) cases, evaluate your database and choose this number.

### Estimating stack usage

For a very large object graph, the depth required to travel the graph for a first search may result in stack overflow.

On Windows operating system, the default stack limit is 1MB; on Solaris it is 8M.

For an object graph whose level exceeds 1000, we recommend that you use level filtering.

### Estimating time

The time required to stream your database depends on your object graphs.

Usually importing is faster than exporting. It takes about 4.5 hours to stream out a 400 MB database with level 1 filtering and 3 hours to stream it in on a sparc Ultra Server.

## Comparison with vimport and vexport

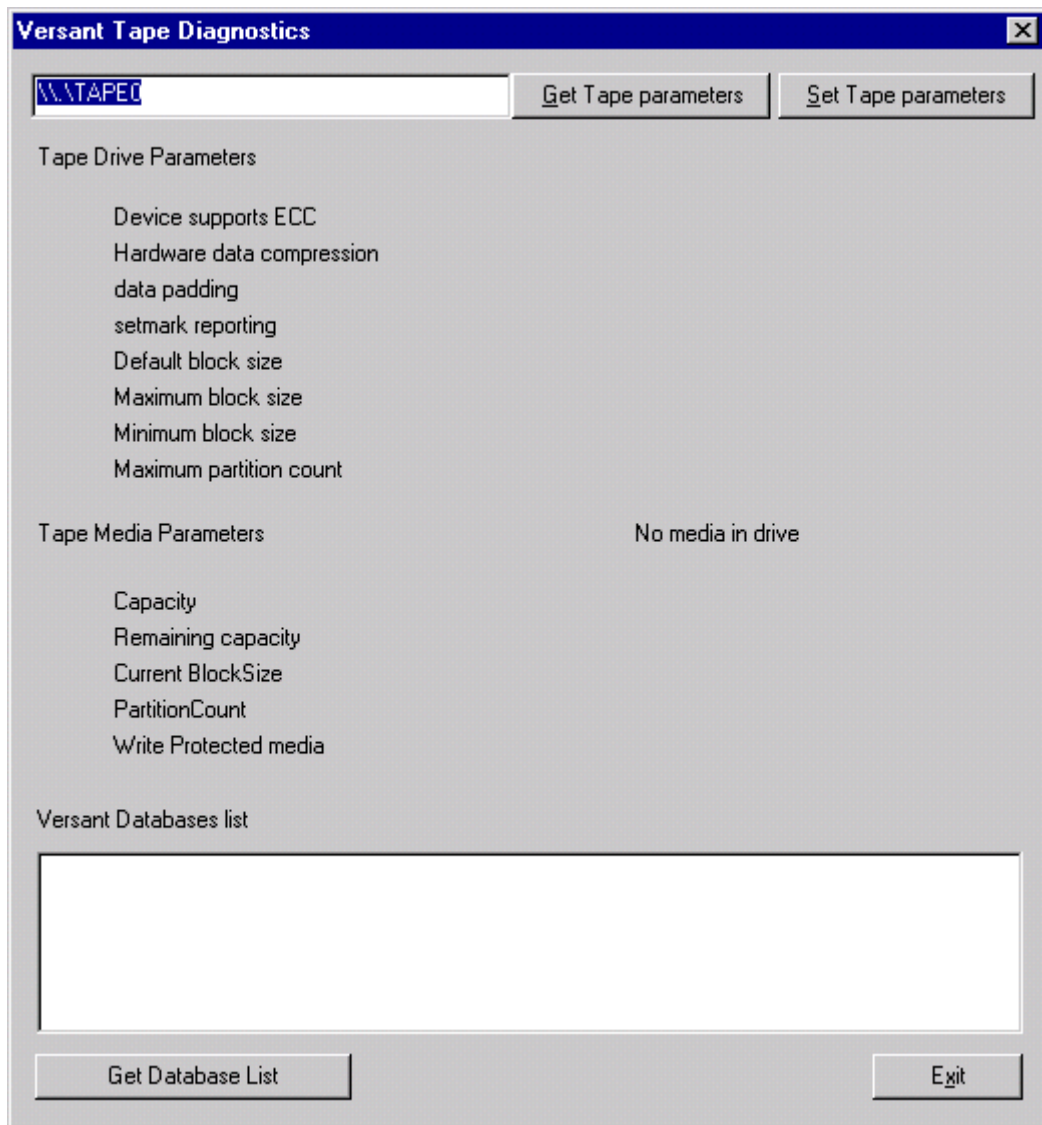
The `vstream` utility replaces the previous utilities `vimport` and `vexport`. The `vstream` has additional functionality and better memory usage.

## vtape

`vtape`

On Windows, you can use the `vtape.exe` utility to view and set SCSI tape drive and media parameters. Following is a screen shot showing the appearance of the `vtape` screen.





The vtape utility is meant to facilitate your use of the vbackup utility.

Tape device names for Windows tape drives are of the form `\\.\TAPE0` - `\\.\TAPEn`. The initial tape drive is `\\.\TAPE0` by default. The `vtape` utility reads and displays the default tape drive's parameters.

To use `vtape`, first invoke it from a command line. Then, to read parameters, in the dialogue box, enter the tape drive name and press the "Get Tape parameters" button. The parameters for `\\.\TAPE` will appear.

If you wish to set tape drive parameters, press the "Set Tape parameters" button. You will then be able to toggle "Hardware data compression" on or off and change the "Current Block Size". If you change the block size setting, make sure that the new value falls between the minimum and maximum block size values for the current tape drive.

The `vtape` utility will also display any Versant databases on the current tape. To view this list, press the "Get Database List" button.

**See also "vbackup" on page 222.**

---

This Chapter explains in details about the Versant Database directories and files, post installation.

The Chapter describes the following:

- Installed Directories and Files
- Other Important Files

## INSTALLED DIRECTORIES AND FILES

The Versant database system physically consists of:

- System files
- Executable utilities
- Header files
- Link libraries
- Class libraries for each language interface
- Application development tools
- At least one database consisting of storage and log volumes, which are files or raw devices
- Database files, including the application process profile file and the server process profile file, that set operating parameters.

## Versant Root Directory Structure

Once the Versant installation is complete, the overall structure of software directories is as given below.

During installation, you specify the location of the `software_root` directory, which can be anywhere you want.

<software\_root>

```

|---ant
|   |---bin
|   |---lib
|---bin
|   |---ansi
|   |---jvi
|   |   |---lp
|   |---memdir
|---demo
|   |---addons
|   |   |---fts
|   |   |---demo
    
```



```

|
|
|
|    |---setcandidate
|    |---setparam
|    |---stringmatch
|    |---tut1
|    |---tutorial
|    |---src
|---jvi
|    |---advanced
|    |    |---DataBaseUtility
|    |    |---MemoryManagement
|    |    |---ObjectMigrationCopying
|    |    |---UserAuthentication
|    |    |---Fund
|    |    |---sharing
|    |    |---event
|    |    |---jca
|    |    |---xa
|    |---IDE
|    |    |---eclipse
|    |---odmg-trans
|    |---query
|    |---tutorial
|    |    |---src
|---vxml
|    |---demo
|---doc
|    |---addons
|    |    |---var
|    |---c
|    |---cxx
|    |---jdo
|    |---jvi
|---h
|    |---cxxcls
|    |---ilang
|    |---odmg
|    |---var
|---jre
|    |---bin
|    |---lib

```

---

```

|---lib
|   |---ansi
|   |---data
|   |---jdo
|   |   |---jca
|   |   |---templates
|   |---jvi
|   |   |---jca
|   |   |   |---jboss
|   |---vxml
|   |   |---dtds
|   |   |   |---v1.0
|   |---eclipse
|   |---schemas
|   |---help_isql
|---uninstaller

```

Following is an explanation of each of these directories:

## Root and System Directories

### Software root directory

<software\_root>

You can choose to place the software root directory and the database root directory anywhere. The default software root directory is:

/usr/local/versant/7.0 (For UNIX)

c:\versant\7\_0\_1 (For Windows)

Once you have specified a location for the software root directory, all other Versant software directories are built from it in a way that cannot vary.

In the following description, all file locations are given relative to this directory. For e.g., the ./h/cxxcls directory is located at <software\_root>/h/cxxcls on Unix and the .\h\cxxcls folder on Windows is located at <software\_root>\h\cxxcls.

## Ant directory

```
.\ant
./ant
```

This directory contains the stripped down Apache Ant 1.5 (minimal configuration).

## bin directory (General executables)

```
./bin
```

This executables directory contains command line utilities for administering Versant and invoking tools. Executables from all the components of Versant Object Database are copied here.

It contains a script `envsettings.csh` for setting environment variables. It also contains a script `jvisetup.csh` for setting up Java Versant Installation, only if Java Versant Installation is chosen and `varsetup.csh` for setting up Versant Asynchronous Replication Installation, only if Versant Asynchronous Replication Installation is chosen.

The bin directory also contains the Batch files and script files for JDO as follows:

File name	Description
<code>jdosetup.csh</code>	To setup the JDO environment
<code>enhance.sh</code>	To enhance the JDO persistence capable files
<code>workbench.sh</code>	To launch the Workbench
<code>schematool.sh</code>	To invoke the schema tool
<code>updatevoa32schema.sh</code>	To invoke the Conversion Tool for VOA 3.2.x customers. Before invoking this tool, please make sure you have used <code>convertldb</code> utility to convert VOA 3.2 x databases to VOD 7.0 x databases.

```
.\bin
```

This executables directory contains command line utilities for administering Versant and invoking tools. Executables and `dll`'s from all the components of Versant Object Database are copied here. It also includes dynamic link libraries for Java Versant Interface (JVI) and C++ debugging libraries for your compiler.

```
.\bin\ansi
./bin/ansi
```



This directory contains the ANSI dynamic link libraries.

The bin directory also contains the Batch files and script files for JDO as follows:

File name	Description
<code>jdosetup.bat</code>	To setup the JDO environment
<code>enhance.bat</code>	To enhance the JDO persistence capable files
<code>workbench.bat</code>	To launch the Workbench
<code>schematool.bat</code>	To invoke the schema tool
<code>updatevoa32schema.bat</code>	To invoke the Conversion Tool for VOA 3.2.x customers. Before invoking this tool, please make sure you have used <code>convertldb</code> utility to convert VOA 3.2 x databases to VOD 7.0 x databases.

The following is the list of launch scripts to be used to invoke various Versant GUI Tools:

GUI Tool Name	Launch Scripts
Administration Console	<code>vdbadmin.bat</code>
Object Inspector	<code>vdbinspector.bat</code>
Monitoring Console	<code>vdbmonitor.bat</code>
Versant Developer Console	<code>vconsole.bat</code>

`.\bin\memdir`  
`./bin/memdir`

This directory contains Versant ReVind(VSQL) related files. The ReVind server and utilities use this directory to create some temporary files.

## demo directory

`.\demo\addons`  
`./demo/addons`

A directory containing subdirectories and the respective demos for all Versant add-on components like FTS, HABACKUP, VAR, Vcompactdb, VSQL and Warm Standby.

`.\demo\c`

`./demo/c`

A directory containing demo and tutorial for C interface.

`.\demo\cxx`

`./demo/cxx`

A directory containing demo and tutorial for C++ interface.

`.\demo\jdo`

`./demo/jdo`

A directory containing demo and tutorial for JDO interface.

`.\demo\jvi`

`./demo/jvi`

A directory containing demo and tutorial for Java Versant interface.

`.\demo\vxml`

`./demo/vxml`

A directory containing demo programs for VXML (Versant XML Toolkit).

## **doc directory**

`.\doc`

`./doc`

A directory containing all the documentation for Versant Object database. This contains the following manuals:

- Versant Database Administration Manual
- Versant Database Fundamentals Manual
- Release Notes for Versant Object Database for 7.0.1.4

`.\doc\addons`

`./doc/addons`

A directory containing documentation for all Versant Add-ons. This directory contains the following Versant Manuals in PDF format:

- 
- Versant Veddind Usage Guide
  - Versant High Availability Backup Usage Manual
  - Versant Vorkout Usage Guide
  - Versant ReVind Reference Guide + Release Notes
  - Versant/ODBC Reference Guide + Release Notes (for Windows platform)
  - Versant Warm Standby Usage Guide

```
.\doc\addons\var
./doc/addons/var
```

This directory contains the following Versant Manuals:

- Versant Asynchronous Replication (VAR) Manual
- VAR C++ Reference Manual (CXXdoc – html format)
- VAR Java Reference Manual (Javadoc – html format)

```
.\doc\c
./doc/c
```

A directory containing documentation for C Versant manuals in PDF format.

- C/Versant Reference Manual

```
.\doc\cxx
./doc/cxx
```

A directory containing documentation for C++ interface. This also contains an introductory demo for C++ interface. This directory contains the following Versant Manuals in PDF format:

- C++/Versant Reference Manual
- C++/Versant Usage Manual

```
.\doc\jdo
./doc/jdo
```

A directory containing documentation for JDO interface. This directory contains the following Versant Manuals:

- Versant JDO Interface Usage Manual (PDF)
- Versant JDO Interface Reference Manual (Javadoc – HTML format)

```
.\doc\jvi  
./doc/jvi
```

A directory containing documentation for Java Versant interface. This also contains an introductory demo for JVI interface. This directory contains the following Versant Manuals:

- Java/Versant Interface Usage Manual (PDF)
- Java/Versant Interface Reference Manual (Javadoc – HTML format)

### **h directory (header files)**

```
.\h  
./h
```

A directory containing the system header files such as `omapi.h` (which contains C/Versant functions).

```
.\h\cxxcls  
./h/cxxcls
```

A directory containing the C++ header files which contain the definitions of the database and library classes that comprise the C++/Versant interface.

```
.\h\cxxcls\template  
./h/cxxcls/template
```

A template directory containing definitions of C++/Versant template classes, such as the dictionary and array types.

```
.\h\odmg  
./h/odmg
```

A ODMG header files directory containing definitions of the database and library classes that comprise the ODMG C++/Versant interface.

```
.\h\ilang  
./h/ilang
```

A language information directory containing files with information about the native language in which you are programming.

---

```
.\h\var
```

```
./h/var
```

A VAR header files directory containing definitions of VAR/C++ API classes.

## jre Directory

```
.\jre
```

```
./jre
```

This directory contains Java Runtime Environment (JRE) required by the GUI tools, VAR and the uninstaller.

```
.\jre\bin
```

```
./jre/bin
```

This directory contains the dlls required for Java application.

```
.\jre\lib
```

```
./jre/lib
```

This directory contains libraries required for Java application.

**NOTE:-** The JRE directory is installed even if you do not select any Java component while installing the Versant Object Database. This is because the installer needs these files to run itself.

## lib directory

```
.\lib
```

```
./lib
```

This library directory contains system libraries and error message files for Versant ODBMS. It also contains the Versant JDO Interface.

```
.\lib\ansi
```

```
./lib/ansi
```

This directory contains ANSI production libraries for Visual Studio .NET 2003 compiler on Windows.

```
.\lib\data
```

```
./lib/data
```

This directory contains data files required for Internationalization support.

```
.\lib\jdo  
./lib/jdo
```

This directory contains data files required for JDO. It also contains the JCA adapter for EJB applications.

```
.\lib\jvi  
./lib/jvi
```

This directory contains data files required for Java Versant Interface and JCA.

```
.\lib\vxml  
./lib/vxml
```

This directory contains data files required for Versant XML Toolkit.

```
.\lib\eclipse  
./lib/eclipse
```

This directory contains the eclipse plugins for editing the mapping in Eclipse 3.0 and 3.1.

```
.\lib\schemas  
./lib/schemas
```

This directory contains Versant ReVind(VSQL) related files. This directory contains 3 schema files which are needed by the schload utility of ReVind.

```
.\lib\help_isql  
./lib/help_isql
```

This directory contains the help files required by the Versant ReVind(VSQL) utilities.

**NOTE:-** Although the ODBMS library files have the same name for all releases, the C++ library files have different names for each compiler.

## uninstaller directory

```
.\uninstaller  
./uninstaller
```

This directory contains the uninstaller and its related files.

---

## Database Directories

During installation, you will be asked for a location for the database root directory. This directory can be anywhere, and you may decide to locate it on another machine acting as a server.

Following are descriptions of the database root directory, and the directories and files associated with each individual database.

### Database root directory

#### **database\_root**

```
.\database_root  
./database_root
```

This is the database root directory.

The database root directory is the root directory for individual database directories created by the database utility `makedb`.

The database root directory can be either local or remote. If a remote directory is used, you must be able to access it with your specification of username and password.

Normally you will create this directory apart from the Versant software root directory, as you will probably want to continue to use the databases that you create even after you install a newer version of Versant at a later time.

The location of the local database root directory must be specified with the `VERSANT_DB` location parameter before you can access, create, or remove a database located under that root directory. The location of a remote database root directory must be specified with the `VERSANT_DB@node` location parameter before you can access, create, or remove a database on a remote machine named `node`.

In (extremely) unusual circumstances, you might want to create multiple database root directories on the same machine. In such a case, you must specify the location of the root directory you want to use with the `VERSANT_DB` and/or `VERSANT_DB@node` location parameters before you can access, create, or remove a database located under that root directory. This practice is not recommended.

The default database root directory is on the local machine under the software root directory.

```
/usr/local/versant/7.0/db  
c:\versant\7.0\db
```

## Database Directory

### **database\_directory**

```
.\database_root\database_directory
./database_root/database_directory
```

A database directory.

Each time you take the first step in creating a database by using the `makedb` utility, a database directory with the same name as the database will be created under the database root directory. Database directories must branch from a database root directory.

For example, if you create a database named `mydb` under the database root directory (as mentioned above), then Versant will create the following database directory:

```
/usr/local/versant/7.0/db/mydb
c:\versant\7.0\db\mydb
```

If a database consists of files, the database directory contains the database volumes for that database.

## Database Volumes and Files

### **For UNIX**

A database on UNIX can use either files or raw devices for the database volumes.

If you are using files, they may be local or remote. If the database volumes are remote UNIX files, they may be accessed through NFS.

**NOTE:-** We strongly discourage you from accessing database files through NFS, because performance will be severely affected.

It is much better to create a database on the host where the file system is physically located and then access it with `VERSANT db@host` protocol rather than with NFS protocol. This allows a Versant server process to directly access the database files.

If you want to use a UNIX raw device for a database volume, make sure that the partition you use does not include cylinder 0 or cylinder 1. When a UNIX partition is used for a raw device, all cylinders allocated to that partition will be used.

**For more information refer to “Database Creation Basics” on page 54, in "Chapter 2 - Database Creation".**



---

## For Windows

A database on Windows can use only files for the database volumes.

## System Volume

### **system**

```
.\database_root\database_directory\system  
./database_root/database_directory/system
```

The database system volume, `system`, is a file or raw device associated with a database that stores class descriptions and object instances.

There is one `system` volume for each database. Additional data volumes can be added to a database to increase capacity.

The system volume for a database is created with the `createdb` utility. If `system` is a file, it is located in the database directory.

## Physical log volume

### **physical.log**

```
.\database_root\database_directory\physical.log  
./database_root/database_directory/physical.log
```

The Physical Log Volume contains physical data information for logging and recovery.

The logical log volume, `logical.log`, and physical log volume, `physical.log`, are files or raw devices associated with a database that record transaction activities and provide information for roll back and recovery.

There is one `physical.log` and one `logical.log` for each database.

The logical and physical log volumes for a database are created with the `createdb` utility. If `physical.log` and `logical.log` are files, they will expand as needed and be located in the database directory.

## Logical log volume

### **logical.log**

```
.\database_root\database_directory\logical.log  
./database_root/database_directory/logical.log
```

The Logical Log volume contains transactions and redo information for logging recovery and rollback.

The logical log volume, `logical.log`, and physical log volume, `physical.log`, are files or raw devices associated with a database that record transaction activities and provide information for roll back and recovery.

There is one `physical.log` and one `logical.log` for each database.

The logical and physical log volumes for a database are created with the `createdb` utility. If `physical.log` and `logical.log` are files, they will expand as needed and be located in the database directory.

## Application Process Profile File

### **database\_name**

```
$HOME/.osc/database_name  
%HOMEDRIVE%%HOMEPATH%\osc\dbname
```

The application process profile file.

A client application process contains the application, part of the Versant Manager software module, a network layer, and an object cache.

When an application starts a session, the operating environment will be set according to the settings in the application process profile file. You can customize the operating environment for an application process by modifying the application process profile. If there is no application process profile file, the system uses defaults.

For any Versant installation, there is one application process profile file per database. When you use either the `makedb` or `makeprofile` utilities, Versant will check the `.osc` (For UNIX) or `osc` (FOR Windows) directory to determine if a file with the same name as the database exists. If an application process profile file exists, Versant will not create a new one. If it does not exist, Versant will create one.

You can use the `makeprofile` utility to create several application process profiles, change their names, edit them, and then specify which one you want to use at run time by changing its name to the database name.

In the application process profile there are parameters that you can specify either before or after a database has been created. These parameters are read each time a session starts. In a session where multiple databases are connected, if you change the parameters in application process profile, the changes will be reflected to other users only when they access the modified application process profile file.

---

Functional parameters in the application process profile file are:

**alias**

Aliases for local and remote databases.

**connect**

Automatic database connections to make each time a session starts.

**max\_processes**

The maximum number of processes within the system.

**signal\_block**

A parameter that will block unwanted signals, such as `Ctrl-C`, that might otherwise damage shared memory.

Tuning parameters in the application process profile file are:

**heap\_size**

A pre-allocation hint for the size of the application heap, which includes cached objects, the cached object descriptor table, and other data structures.

**swap\_threshold**

Swap threshold. If less than this amount of the application heap is used, object swapping will not be attempted by the system.

**For more information refer to “swap\_threshold” on page 82, in "Chapter 3 - Database Profiles".**

## LOGFILE

```
.\database_root\database_directory\LOGFILE  
./database_root/database_directory/LOGFILE
```

This file logs database and system messages and errors. The location of this file can be configured using the server profile parameter `versant_be_logfile`. By default, error logging is enabled.

In addition to error number and error message, entries in the error log file also includes context information, such as operating system error number, file name and so on. The exact information varies from error to error.

The following message will be logged in the `LOGFILE` if the database volumes are running out of space:

```
"Process:<Process-id> <Date(includes time)> <msg>
```

## DBA Utility Logging

This `LOGFILE` is also used as the default file to log all DBA utility activities. If you specify `versant_be_dbaloggingfile` parameter or use the utility

```
dbtool -dbalog -setfile
```

option to specify any other file, then the logging will occur there. Failing that, if `versant_be_logfile` is specified, it will be logged there else in the `LOGFILE`.

The general format of the messages that will be logged into the file used for logging DBA activities is:

```
<PID:#####>; <HOST: hostname>
<START | END | INVOKED>; <Complete command line>;
<Utility dependent extra information like "invoked
directory", effective user ID>
```

Each entry in `LOGFILE` will have one or more of the following fields:

**PID:** Process ID of the utility and not that of the process, which is logging this message into the file.

**HOST:** Hostname of the machine where the utility is run.

**START:** Indicates the start of the utility. On successful completion there will be a corresponding **END** entry.

**END:** Indicates the completion of the utility. This entry might include an error (see `ERRCODE` field). If a utility has run successfully, the **START** and **END** entries have to match up as pairs. This also means that if there is a **START** entry but no **END** entry, the utility has either not finished doing its job, is hung or has been killed with a signal that cannot be caught.

**INVOKED:** Indicates the invocation of the utility. This field and the combination of **START** and **END** fields are mutually exclusive.

---

**ERRCODE:** This may appear as part of the `END` entry. It indicates with what error the utility failed. If the utility succeeded, this value would be zero. If the error code is difficult/impossible to obtain, this field would not be present.

**CWD:** Directory from where the utility was invoked.

**RUID:** Real user ID. On platforms where this field is irrelevant, this field will not be present or it will have the string “N/A”.

**EUID:** Effective user ID. On platforms where this field is irrelevant, this field will not be present or it will have the string “N/A”.

Here are a few examples of entries in the per-database configurable file:

```
PID: 12345; HOST: mymachine; Wed Feb 7 12:56:23 2001
START; vbackup -dev /vol01/backup/dbname.020701.level0
-backup dbname; CWD: /vol03/versant;
RUID: srao; EUID: srao;
```

```
PID: 12346; HOST: ourmachine; Wed Feb 7 12:59:23 2001
START; vbackup -dev rf.0 -log dbname@yourmachine;
CWD: /vol03/versant; RUID: srao; EUID: srao;
```

```
PID: 12345; HOST: hismachine; Wed Feb 7 19:37:23 2001
END; vbackup -dev /vol01/backup/dbname.081547.
level0 -backup dbname; CWD: /vol03/versant;
RUID: srao; EUID: srao;
```

```
PID: 12346; HOST: hermachine; Wed Feb 7 12:59:23 2001
END; vbackup -dev rf.0 -log dbname@yourmachine;
CWD: /vol03/versant; RUID: srao; EUID: srao;
```

```
PID: 12347; HOST: yourmachine; Wed Feb 7 12:56:23 2001
INVOKED; dbuser -add -n smartperson -defaultpasswd
dbname@remotehost; RUID: srao; EUID: srao;
```

```
PID: 13567; HOST: theirmachine; Wed Feb 7 12:56:23 2001
INVOKED; dbuser -chpasswd -n username -opasswd
<notdisplayed> -passwd <notdisplayed>
dbname@localhost; RUID: srao; EUID: srao;
```

## Server process profile

### `profile.be`

```
.\database_root\database_directory\profile.be  
./database_root/database_directory/profile.be
```

The database server process profile file.

When a database starts up, the database server process reads the server process profile to determine the location of the database volumes and to set the database operating environment. If a server process profile does not exist for a database, you cannot start that database.

The server process profile file is named `profile.be` and is located in the directory for the database, which must branch from the database root directory.

If `profile.be` does not exist, you cannot start the database.

The server process profile is created with the `makedb` utility.

The server process profile contains several kinds of parameters. Some are used only when a database is created, and others are used each time a database is started.

**For more information, please refer to Chapter 3 “Database Profiles” on page 71.**

## Lock file

### `.lock`

```
./database_root/database_directory/.lock
```

### `lock`

```
.\database_root\database_directory\lock
```

This file contains database state information and is created when you create a database with the `makedb` utility. Versant will use the `.lock` file to determine whether a database has already been started. If the `.lock` (For UNIX) or `lock` (For Windows) file is missing, you will not be able to start or recreate the database.

If the `.lock` (For UNIX) or `lock` (For Windows) file is destroyed for some reason, use the `dbinfo` utility to create a new one:

1. Change directory to the appropriate database directory,
2. Run the `dbinfo` utility with the following arguments:

```
dbinfo -c database_name
```

---

## Password file

**.pw**  
./database\_root/database\_directory/.pw

**pw**  
.\database\_root\database\_directory\pw

This file `.pw` (For UNIX) or `pw` (For Windows) is created only if the DBA authentication is done using a password-based mechanism. The specified password is stored in this file and is accessed everytime a utility that requires DBA authentication is invoked like `createdb`, `dbinfo`, `vmovedb` etc.

If DBA forgets the password, the DBA can login as an OS user, delete the password file and recreate a new password using the `dbuser` utility.

**For more information refer to “dbuser” on page 182, in “Chapter 4 - Database Utilities”.**

**CAUTION:-** The DBA authentication will not be performed in the absence of this file under any circumstances.

## Backup File

**.vbackup**  
./database\_root/database\_directory/.vbackup

**vbackup**  
.\database\_root\database\_directory\vbackup

**.vbackup**

This file `.vbackup` (For UNIX) or `vbackup` (For Windows) contains data necessary for reporting `vbackup` progress. Any attempts by the user to tamper with the `.vbackup` file will result in incorrect progress bar of `vbackup`.

## Shared Memory File

**.sharemem**  
./database\_root/database\_directory/.sharemem

### **sharemem**

`.\database_root\database_directory\sharemem`

When you create a database with the `createdb` utility, Versant will create a file named `.sharemem` in your individual database directory. The contents of the `.sharemem` (For UNIX) or `sharemem` (For Windows) file in a database directory are used as a key to get a unique id for the shared memory of the database server process.

The `createdb` utility creates this file. The `removedb` utility removes this file. You should never remove the `.sharemem` (For UNIX) or `sharemem` (For Windows) file yourself, because if this file is missing, you will not be able to start or recreate the database.

If the `.sharemem` (For UNIX) or `sharemem` (For Windows) file is removed, in some cases shared memory used by the database will exist, but the `stopdb`, `startdb`, and `removedb` utilities cannot discover it. Then, when you recreate the database, a `.sharemem` (For UNIX) or `sharemem` (For Windows) file may have the same memory identifier, which causes problems when you try to start the new database.

If the file is removed, do the following:

### **For UNIX**

1. Remove outstanding shared memory associated with this database using `ipcrm`.
2. Change to the database directory.
3. Create an empty `.sharemem` file in the database directory with `touch`.

### **For Windows**

The workaround is to create an empty `sharemem` file in the database root directory.

After creating the empty `.sharemem` file, you can then start the database with the `startdb` utility or, if it has been removed with `removedb`, you can recreate it with the `createdb` utility.

## **Database type file**

### **.dbtype**

`./database_root/database_directory/.dbtype`

### **Personal.flg or Group.flg**

`.\database_root\database_directory\Personal.flg` or  
`.\database_root\database_directory\Group.flg`



---

When you create an individual database directory with the `makeprofile` or `makedb` utility, Versant will create a file named `.dbtype` in your individual database directory.

The contents of the `.dbtype` (For UNIX) file in the database directory indicate the database type, whether personal or group. On Windows, `.dbtype` does not exist. Instead, there is either a text file called `Personal.flg` or `Group.flg` that contains the database owner's user name. This file is created when you create a database with the `makedb` utility.

If the file is missing, the database can't be created or started.

### For UNIX

The permission bits on the `.dbtype` file establish the database type: `0600` means a personal database and `0660` indicates a group database. If the file is destroyed, recreate a hidden file in the database directory.

### For Windows

If the file is destroyed, recreate a hidden file with the appropriate name in the database directory that contains the database owner's user name.

## Trace Log File

```
.systrace  
./database_root/database_directory/.systrace  
systrace  
.\database_root\database_directory\systrace
```

This file contains the trace log when tracing is turned on. If tracing is not on, this file will contain "error messages". Some of these "error messages" may not really be errors, because they may have been handled at a higher level.

For more information refer to "dbtool" on page 155, in "Chapter 4 - Database Utilities".

## Configuration Files

During installation, Versant will create two configuration files: a machine configuration file (`.oscxyyzz` or `vrxyyzz.ini`) and a system information file (`sysinfo`) depending on your Operating System. The contents of the machine and system information files will vary depending upon the type of installation you perform.

For more information, on Setting up Environment Parameters, please refer to the *Release Notes*.

## Machine configuration file

```
.oscxyyz  
vrxyyz.ini  
/etc/.oscxyyz  
%HOMEDRIVE%\WINNT\vrxyyz.ini
```

If you install using the installation program, the installation program will create a machine configuration file in your `/etc` directory.

The machine configuration file, where `xx` is the major release number, `yy` is the maintenance release number, and `zz` is the minor release number. The contents of the file at least have the information of the following parameters:

```
VERSANT_ROOT  
VERSANT_DBID_NODE  
VERSANT_DBID  
VERSANT_DB
```

For example, for Release 7.0.1.3 the machine configuration file is:

```
/etc/.osc070001  
c:\WINNT\vr070001.ini
```

## System information file

```
sysinfo  
./lib/sysinfo  
.\lib\sysinfo
```

The installation program will create a system information file under your `<software_root>/lib` directory. The location of the `lib` directory containing the `sysinfo` file depends upon your choice of software root.

For example, for a default software root the location of the file will be:

```
/usr/local/versant/7.0/lib/sysinfo  
c:\versant\7.0\lib\sysinfo
```

---

## Statistics File

**.vstatsrc**

`./lib/.vstatsrc`

**vstats.ini**

`.\lib\vstats.ini`

When it starts, `vstats` will read a configuration file in which derived statistics can be pre-defined and named. These files can contain comments beginning with `#` and commands to define new derived statistics.

Each definition of a derived statistic must appear on a separate line and should have the following syntax:

```
define stat_name [ args ] = expression
```

Elements of a definition line are:

**stat\_name**

Name of the new statistic.

**args**

A space delimited list of arguments.

**expression**

The expression that creates the derived statistic.

In a `.vstatsrc` file, each definition must be on a single line ended with a hard return. Definitions cannot be continued to a next line by using a `\` symbol, although visually they can wrap to a hard return.

Statistic names must be specified as shown in the *Versant Database Fundamentals Manual*, except without the `STAT_` prefix and transposed to lower case.

**For more information on Statistic names, please refer to the chapter “Statistic Collection” in the *Versant Database Fundamentals Manual*.**

For connection and database statistics (those with the prefixes `be_` and `db_`), you must also use a database name since the same statistics may be available for multiple databases. The general syntax is:

```
'statistic_name database_name'
```

You can use simple mathematical formulas to combine and modify statistics to produce a derived statistic. For example:

```
vstats -stats se_reads + se_writes
```

In these formulas you can use the standard numerical operators `+`, `-`, `*`, `/`, `(`, and `)`. You can also use the following special functions:

### **delta**

The difference between an expression's current value and it's next most recent value.

### **min**

The minimum value of an expression over the `vstats` run.

### **max**

The maximum value of an expression over the `vstats` run.

### **avg**

The average value of an expression over the `vstats` run.

In addition to being useful for defining new statistics, you can also define aliases. For example:

```
define objs_sec = db_objs_sec database1  
define o = objs_sec
```

Once a statistic has been defined, you can use it in `vstats` commands and options just as if it were a pre-defined Versant statistic. Statistics defined in either of the two `.vstatsrc` files will be listed when the `-summary` command is invoked.

For example:

```
define se_hit_ratio = se_locates/(se_locates+se_reads)  
define db_hit_ratio db = db_data_bytes_located db /  
(db_data_bytes_located db +db_data_bytes_read db)  
define avg3 n1 n2 n3 = (n1+n2+n3)/3
```

### **For UNIX**

A system-wide `.vstatsrc` file in the `VERSANT lib` directory will always be read. If a file named `.vstatsrc` exists in your home directory, it will also be read.

---

For more information on Statistics collection and its usage, please refer to the chapter “Statistic Collection” in the *Versant Database Fundamental Manual*.

## OTHER IMPORTANT FILES

### Database System Identifier File (osc-dbid)

The database system identifier file is a special file that contains information about all databases in a network of databases.

A database system identifier file must exist and be accessible to your machine before you can create a database.

Normally, the database system identifier file will be on a machine acting as a server and will have already been created by the time you perform an installation.

A Versant database system is a group of databases among which you can safely connect and share objects.

Versant also implements the concept of distributed database network systems in which:

- Any database in a system can safely interact with any other database in that system,
- Any database in a system can operate while completely disconnected from all network communications,
- Any number of databases can belong to a particular network system,
- Databases in a particular system can be located anywhere in a network,
- Any number of systems can be created.

### Logical object identifier - LOID

As per the object model, when you create an object, it is assigned a unique identifier number called the "loid".

For objects to be safely shared among databases, this number:

- Must be unique within a particular system of databases,
- Can never be reused even if an object is deleted,
- Can never be changed even when an object is moved.

Due to these properties, this number can be used as a persistent link to an object regardless of its database or memory location.

To ensure that the logical object identifier for each object is unique among all objects in all databases in a system of databases, Versant object identifiers are composed of two parts:

- A unique-within-the-database number, and
- A unique-among-other-databases number.

A database does not have to communicate with other databases to ensure the uniqueness of the part of the object identifier that is unique within itself: it just needs to keep track of numbers already used.

To assign an object the part of the object identifier that is unique to the database itself, a database does have to communicate with other databases at least once, at the time it is created. Once a database has been created, it can then store its identifier number and no further communication with other databases is necessary to create objects.

So that Versant does not have to communicate with all existing databases each time a new database is created, a database system identifier file is used. The name of this file is `osc-dbid`, and this file contains information about each database in the system that it defines, including its identifier number.

When you create a database, Versant finds the `osc-dbid` file, reads it, assigns the next available identifier number, and updates it with information about the new database. Versant also uses the `osc-dbid` file whenever you remove a database or request information about all databases in the system.

During installation, you will be asked for the machine and path of the `osc-dbid` file for the database system that you want the new installation to join.

Typically, you will want to place the `osc-dbid` file not on the local machine but rather in a visible directory on a machine that you designate as a network server, because it must be able to be found by all machines in the network. Only if you have a standalone installation that will never use objects in databases on other machines, you can place the `osc-dbid` file on your local machine.

If you want your installation to join a system of distributed database, we strongly recommend that you install first on the machine that will act as your database system host, create a database system file at that location, and then specify the machine and path of that file when you install on each client machine. Installing first on the database system host will ensure that when you install on a client machine you know the location of your database system file and do not have to make corrections later.

After a network of databases has been defined, you can move the location of the `osc-dbid` file. However, this is not recommended, because this would require your going to all machines in the network and revising the Versant environment variables and configuration files that point to the old location.

In any case, before you can create a database:

1. an `osc-dbid` database system file must exist at a location accessible to your machine, and

2. the location of the `osc-dbid` file must have been specified to your machine.

Since the `osc-dbid` database system identifier file is only used when databases are created, removed, and listed, you can connect to a database even when the node containing the `osc-dbid` file is down. However, since the `osc-dbid` file is not read each time you make a database connection, you could conceivably create simultaneous connections to databases belonging to differing database network systems.

**CAUTION:-** You must avoid doing this, because databases in differing database network systems could have the same identifier number, and thus there could be objects in the differing systems with the same logical object identifier.

If the `osc-dbid` database system file for a distributed database network system is destroyed, you must recover the `osc-dbid` file from a backup before you can create new databases, remove existing databases, or list the databases in that system.

If you have no `osc-dbid` file backup, recovery procedures are complex. Although you can use the `dbid` utility to create a new `osc-dbid` file, creating a new `osc-dbid` file means that identifiers for new databases will probably duplicate the identifiers for existing databases. This means that you must create a new network of databases, and then create new objects in the new databases that are copies of the old objects (you must be careful not to migrate the old objects).

If your `osc-dbid` file is destroyed, please call Versant Customer Support.

The default location for the `osc-dbid` file on a UNIX installation is the `/home` directory on the local machine. The default location for the `osc-dbid` file on a personal computer installation is the database root directory on the local machine.

**The following explains what happens after the installation program asks for a path to an `osc-dbid` file:**

### No path

If you do not specify an `osc-dbid` machine and directory, the installation program will create a new `osc-dbid` on your local machine in your home directory.

### Same machine

If you specify that the `osc-dbid` machine is the same as the installation machine, the installation program will first look for an existing `osc-dbid` file in your home directory.



---

If an `osc-dbid` file already exists in the specified directory, you will join the existing system of databases.

If an `osc-dbid` file does not exist in the specified directory, the installation program will create an `osc-dbid` file in the specified directory and a new database system will be started.

An `osc-dbid` file may not exist in the specified directory because this is the first installation of Versant on this machine or because you want this installation to create a new database system.

### Remote machine

If you specify that the `osc-dbid` machine is different than the installation machine, the installation program will not look to confirm that an `osc-dbid` file exists in the specified directory and will not create a new `osc-dbid` file.

In all cases, the installation program will record the path of the new or existing `osc-dbid` file in the system information file `sysinfo` file if your installation will use local files and in the machine configuration file `.oscxyyzz` if your installation will use remote files.

### See also “Configuration Files” on page 303.

Because Versant does not check for the existence of an `osc-dbid` file if you specify a remote machine, during installation you do not have to be connected to a network and the first installation of Versant does not have to be on the machine that will eventually contain the `osc-dbid` file.

However, we recommend that the first installation of Versant be on the machine that will contain the `osc-dbid` file in order to reduce the chance of an unintended outcome.

For example, if you specify a remote path to `osc-dbid` and then later decide on a different path for `osc-dbid`, then you will not be able to create a database until you manually specify the correct path on each and every installation already made.

Or, if you do not specify a path to `osc-dbid` at all, there is the danger that before you get a chance to tell the local Versant installation about the remote `osc-dbid` file, you will create a database using the local `osc-dbid` file, which breaks the object model because unique object identifiers across the system of databases cannot be guaranteed.

Before installation, you can find the machine and path of the database system file by running the `oscp` system utility of an installation whose database system you want to join.

Or, if Versant files are local, you can inspect the local `/db/sysinfo` system configuration file. If you are using Versant files on a remote machine, you can inspect your `/etc/.oscxyyzz` machine configuration file. In a configuration file, the machine containing the `osc-dbid` file will be associated with the entry for `VERSANT_DBID_NODE`, and the directory will be associated with the entry for `VERSANT_DBID`.

For example, if `osc-dbid` is on a machine named `server_machine` and in the directory `/visible_directory`, you would see the following entries in a configuration file:

```
VERSANT_DBID_NODE    server_machine
VERSANT_DBID         /visible_directory
```

## License File (license.xml)

The license file is a special file containing licenses for Versant components.

A license file must be present on each server in the network. This is a text file with XML-like tags.

The default license file is located at the installation directory and has the name “`license.xml`”. Thus the default license file for a Versant installation would be at `$VERSANT_ROOT/license.xml`.

It is possible to specify a different license file for the installation. In this case the environment variable `VERSANT_LICENSE_FILE` should be set to point to the license file. This environment variable needs to be set for the backend server (for `ss.d` and `VERSANTD.EXE`) also.

The license file consists of specific tags and values. An example of the license file follows.

```
<versant-license>
  <component>
    <component-name>ODBMS</component-name>
    <component-version>7.0</component-version>
    <product-name>VERSANT</product-name>
    <license-type>Test</license-type>
    <hostid>808046c4</hostid>
    <customer>ABC Corp</customer>
    <contact-name>XYZ</contact-name>
    <contact-email>xyz@abc.com</contact-email>
    <component-usage>ODBMS license</component-usage>
    <component-key>281B009901F800B206224003</component-key>
    <expiry-date>06/30/2002</expiry-date>
    <key-date>01/16/2002</key-date>
  </component>
</versant-license>
```

---

The description of the various tags in the xml file is given below:

No.	Keyword	Description
1	versant-license	Encloses all licensed versant products. May occur multiple times in each license file
2	component	Each licensed component should be enclosed within this tag
3	component-name	Licensed component name
4	component-version	Version number (most two significant digits) of the component e.g. 6.0
5	product-name	Name of product
6	license-type	Type of environment for the product
7	hostid ipv4 ipv6	Identification of the machine. Only one of the three keywords is allowed.
8	customer	Customer name.
9	contact-name	Name of the person requesting license.
10	contact-email	Email address of the person requesting license. This is the address where the license will be mailed.
11	component-usage	A string for use by the component.
12	component-key	The license key.
13	expiry-date	Expiry date of the license.
14	key-date	Date on which license was generated.

The license is node-locked and any change in the value strings would invalidate the license causing the product to stop working.

The first license, matching the following criteria, is used for performing the license check:

- Component name
- Component version
- Machine identification

Hence a new license needs to be inserted at the beginning of the license file. This will avoid the older license (if one exists) to be picked up for the license checks. The same license file can obtain licenses for different products and/or different versions of the same products and/or for different servers.

The license will be generated by Versant and normally sent to the customer-requesting license through e-mail.

---

This Chapter explains about the “Configuration Parameters”.

The Chapter describes the following:

- Configuration Parameters Overview
- Configuration Parameters for UNIX
- Configuration Parameters for Windows

## OVERVIEW

Configuration parameters enables Versant to:

- Find software files and databases as needed.
- Know where you want to create or remove a particular database.
- Modify or list information about all databases in a particular system of distributed databases.

These parameters are already set by Versant which can be used in most of the cases.

You might only need this Chapter, if you would want to change or set the configuration parameters needed by Versant.

After installation, you only need to append the location of Versant executables to your existing `PATH` environment variable. (On personal computers, the installation program will do this for you.)

**You must explicitly specify or change configuration parameters only in the following situations:**

- If you did not answer all prompts during installation or if you made a mistake.
- If you want to move software or databases.  
As your hardware changes, you might want to move Versant software and/or databases. After you move a database or Versant software, you need to tell Versant the new locations.
- If you want to create multiple, distinct database systems.
- If you are embedding Versant in another application, you might want to create multiple, distinct database systems so that your application need not know about any other Versant databases on the network except those related to its purposes. To do this, you must be sure that data will never be shared among the different systems of databases.
- If you want to create multiple sets of databases within the same database system.  
You might want to create multiple sets of databases managed by the same machine, but branching from different database root directories. This might let you group certain types of databases on different storage devices and would allow you to let certain users "see" only the databases relevant to their tasks.
- If you want to install and selectively use multiple Versant releases on the same machine.

---

In a development environment, you might want to install multiple Versant releases on the same machine and then shift among the releases to confirm the compatibility of various applications and databases.

- If you want to use Versant software located on a remote machine.

You might want to develop an application using a local installation of Versant and later run it using a remote installation of Versant in order to take advantage of a faster machine.

You can also use mechanisms other than configuration parameters to customize your environment. For example, you can set application and database operating parameters by editing your application and server process profile files.

**For more information, on Operating parameters, please refer to Chapter 3 “Database Profiles” on page 71.**

## CONFIGURATION PARAMETERS FOR UNIX

On UNIX systems, Versant does not depend on environment variables, although they optionally can be used to define the Versant environment.

A normal installation of Versant creates two configuration files, one in the machine configuration file, `.oscxyyz`, and one in the system information file, `sysinfo`. The combination of these two files after installation completely defines the Versant environment. The only additional environmental change made during installation is to place the `/bin` directory on your path.

When specifying locations in Versant parameters, you can use NFS mountings.

Besides these Versant parameters, you should also add the Versant executables directory to your machine's existing `PATH` environment variable.

## UNIX Configuration Parameters Location

On UNIX installations you can set configuration parameters in the following places:

### Environment variables

On a per process basis, you can specify paths with environment variables. Both client and server processes will look for environment variables.

### User configuration file

On a per user basis, you can specify paths by creating a user configuration file. Only client processes will look for a user configuration file.

### Machine configuration file

On a per machine basis, you can use or modify the machine configuration file created by Versant during installation. Both client and server processes will look for a machine configuration file.

### System information file

On a per installation basis, you can use or modify the system information file created by Versant during installation. Both client and server processes will look for a system information file.



---

When you install Versant on a UNIX machine, all configuration parameters needed by Versant are recorded either in a machine configuration file in your `/etc` directory or in a system information file in your `VERSANT lib` directory. The contents of these files depend upon how you installed.

You can install Versant on a machine in two different ways:

**Use files on local machine:** You can install Versant so that it uses local Versant files by invoking the installation program with the command `oinstall`.

**Use files on remote machine:** You can install Versant so that one machine uses Versant files on another machine by invoking the installation program with the command `oinstall -f`.

Either way that you install, Versant will create two configuration files:

### Machine configuration file

The directory and name of the machine configuration file is:

```
/etc/.oscxyyzz
```

Where:

`xx` is the major release number

`yy` is the maintenance release number

`zz` is the minor release number

For example for Release 7.0.1.3 machine configuration file is:

```
/etc/.osc070001
```

### System information file

The directory and name of the system information file is `sysinfo` and branches from your `lib` directory:

```
/root/lib/sysinfo
```

The contents of the machine and system information files will vary depending upon the type of installation you perform.

The contents of the file `.oscxyyzz` at least have the information of the following parameters:

`VERSANT_ROOT`

`VERSANT_DBID_NODE`

VERSANT\_DBID

VERSANT\_DB

**For more information, on the location and nature of the machine configuration and system information files, please refer to Chapter 4 “*Directories and Files*” on page 281.**

## UNIX Configuration Parameter Considerations

There are two considerations in choosing where to set parameters: the order in which Versant searches and the precedence of passed parameters.

### Search order

To find the information it needs on each UNIX node, Versant searches environment variables and information files in a predetermined, hierarchical order. If conflicting information is found in the search, the order of the search resolves the conflict: the first value found is the value that is used.

To get needed information on a UNIX node, Versant searches in the following order:

1. Environment variables — per process
2. User configuration file — per user
3. Machine configuration file — per machine
4. System information file — per installation

Since Versant searches in a hierarchical order, you can supply most information on a per process, per user, per machine, or per installation basis.

For example, if you normally create and use local databases in the directory `/usr/local/versant/db`, then you can define the following in your machine configuration file (which is the default location):

```
VERSANT_DB /usr/local/versant/db
```

Then, if you have an application that embeds Versant and works solely with drawings stored in databases in the directory `/myapplication/db`, before starting Versant you can define the following environment variable to override the setting in the machine configuration file:

```
VERSANT_DB /myapplication/db
```

On personal computer nodes, Versant makes a similar search, although the details differ.

---

See also “Configuration Parameters for Windows” on page 351.

### Passing parameters to remote machines

To specify a command or configuration parameter on a remote machine, you can use `@node` syntax. For example, to create a database on a remote machine named `server`, you would use the following `@node` syntax:

```
createdb db1@server
```

Or, to look for an `osc-dbid` file on a remote machine:

```
VERSANT_DBID_NODE@server
```

If you use `@node` syntax, all parameters are sent to the server on `node` (local or remote).

If you do not use `@node` syntax, all parameters, except for `versant_root` and `versant_rel`, are sent to the local server. This means that to send `versant_root` or `versant_rel` to the local server, you must use `@node` syntax.

Configuration parameters explicitly passed to a remote machine override those already set on the remote machine. You should pass parameters only if you want to override the values set on the remote machine.

For example, suppose the UNIX machine `server` has the following environment parameter set:

```
VERSANT_DB /usr/local/versant/db
```

To create a database on `server` in a different directory, you could set the following:

```
VERSANT_DB@server /db
```

Then, if you invoke the create database utility as `createdb@server`, the value of your environment parameter `VERSANT_DB@server` will be passed to the machine named `server` and the database will be created in a directory branching from `/db` rather than `/usr/local/versant/db`.

You can use `@node` syntax for any parameter and set any parameter for remote use in either a configuration file or in an environment variable.

If you doubly define a parameter for the local machine by defining it both with and without `@node` syntax, Versant will look first for a value in the parameter without `@node` syntax and then look for it with `@node` syntax. The first value found will be the value used.

For example, if you are on machine `alpha` and have defined both `VERSANT_DB` and `VERSANT_DB@alpha`, then Versant will use the value in `VERSANT_DB`. However, if you have not defined `VERSANT_DB`, then Versant will use the value in `VERSANT_DB@alpha`.

In any case, you must, of course, specify parameters to be passed in the syntax appropriate to the remote machine. For example:

```
VERSANT_DB@unix_server /usr/local/versant/db
VERSANT_DB@nt_server \versant\db
```

## UNIX Configuration Parameter Procedures

The procedure for setting the Configuration Parameters is:

### Environment variable

To set a configuration parameter in an environment variable, use the name of the environment parameter followed by its value.

For example, suppose you want to set the local database root directory on Solaris environment to be the default location then:

```
VERSANT_DB /usr/local/versant/db
```

If you use C Shell, to set `VERSANT_DB` in your `.login` file in your home directory:

```
setenv VERSANT_DB /usr/local/versant/db
```

If you use Bourne shell, to set `VERSANT_DB` in the `.profile` file in your home directory:

```
VERSANT_DB = /usr/local/versant/db
export VERSANT_DB
```

### User configuration file

You can specify the locations of the software and database directories in a user configuration file that you create.

The format for setting the user configuration file is giving the name of the environment parameter followed by the value.

After creating the user configuration file, use the following environment variable to tell Versant about the file:

```
VERSANT_CFG
```

The name and location of the user configuration file containing specifications for software and database root directories.

---

Only the following VERSANT environment variables can be set in the VERSANT\_CFG file:

VERSANT\_ROOT, VERSANT\_DB, VERSANT\_DBID and VERSANT\_DBID\_NODE.

All other environment variables need to be set outside VERSANT\_CFG.

For example, if you have created a user configuration file name `myconfig.txt` in `/usr/local/versant`, to set the local database root directory on Solaris environment to be the default location, create the following line:

```
VERSANT_DB /usr/local/versant/db
```

Then, to set VERSANT\_CFG to the location of your configuration file, if you use C Shell:

```
setenv VERSANT_CFG /usr/local/versant/myconfig.txt
```

If you use Bourne shell:

```
VERSANT_CFG = /usr/local/versant/myconfig.txt
export VERSANT_CFG
```

## Machine configuration file

In the Versant machine configuration file `/etc/.oscxyyz`, you can create or modify a line for an environment parameter.

The format for lines in the machine configuration file is the name of the environment parameter followed by the value.

For example, to set the local database root directory for Release 7.0.1.3 installed on Solaris environment to be the default location, create or modify the following line in `/etc/.osc070001`:

```
VERSANT_DB /usr/local/versant/db
```

You can make comments in the `.oscxyyz` file on any line in which either the first column is marked with the `#` sign or after a `#` sign on a line containing an entry.

For example, for release 7.0.1.3, local machine `mymachine`, software root path `/usr/local/versant`, and database root path `/db`, the contents of `/etc/.osc070001` could be:

```
# the following line specifies software root
VERSANT_ROOT /usr/local/versant
VERSANT_DB /db #this line is the database root
```

## System information file

Versant system information file `sysinfo` is in your `lib` directory, where you can create or modify a line for an environment parameter.

To set an environment parameter in the Versant `sysinfo` file, use the name of the environment parameter followed by its value.

For example, to set the local database root directory on Solaris to be the default location, create or modify the following line:

```
VERSANT_DB /usr/local/versant/db
```

You should not set the location of `VERSANT_ROOT` in the `sysinfo` file, as Versant needs to know its software directory in order to find the `sysinfo` file in the first place.

You can make comments in the `sysinfo` file on any line in which either the first column is marked with the `#` sign or after a `#` sign on a line containing an entry.

## UNIX Configuration Parameters - Mandatory

When you install Versant, all configuration parameters needed by Versant are recorded in either in the machine configuration file `/etc/.oscxyyzz` or in the system information file `lib/sysinfo`.

After installation, the only thing that you should do is add the Versant executables directory to your machine's existing environment variable `PATH`.

**For more information, on Versant file structures, please refer to Chapter 4 “Directories and Files” on page 281.**

### PATH

The parameter `Path` sets the Executables directory. This is always needed.

After installation, append the location of the Versant executables directory `bin` to your `PATH` environment variable. This will enable Versant to find the executables regardless of your current directory.

For a standard installation, the Versant executables directory will be named `/bin` and be located below the platform directory. For example on Solaris environment the location will be:

```
/usr/local/versant/bin
```

---

If you use C Shell, the search path should be in the `.login` file in your home directory. To amend it to include the Versant `/bin` directory, use the `set path` command.

For example, for a default installation on Solaris (32 bit) environment:

```
set path = (/usr/local/versant/bin $path)
```

If you use Bourne shell, the search path should be in the `.profile` file in your home directory.

For example, to amend it to include the Versant `/bin` directory for a default installation for Solaris (32 bit):

```
PATH=/usr/local/versant/bin:${PATH}
export PATH
```

## UNIX Configuration Parameters - To Be Set In Special Cases

The following configuration parameters are necessary for Versant to run in a standard environment.

You must explicitly set them only in the following situations:

- If you did not answer all prompts during installation.
- If you made a mistake during installation.
- If you have moved your software, database directory, or database system identifier file.
- If you have created a user configuration file.

### VERSANT\_ROOT

The parameter `VERSANT_ROOT` sets the software root directory. This is a Mandatory Parameter.

This parameter enables Versant to find its own files, because they are placed in predetermined places in directories built from the software root directory. On UNIX installations, the default value of `VERSANT_ROOT` is `/usr/local/versant`.

You may set the parameter `VERSANT_ROOT` in any of the following places:

#### VERSANT\_ROOT

In the environment variable `VERSANT_ROOT`.

### **user.cfg**

In a user configuration file, if you also set `VERSANT_CFG`.

### **/etc/.oscxyyyzz**

In the machine configuration file, which is the default location.

You cannot set `VERSANT_ROOT` in the system information file.

For example, if you use C Shell and set `VERSANT_ROOT` in your `.login` file in your home directory:

```
setenv VERSANT_ROOT /usr/local/versant
```

For example, if you use Bourne shell and set `VERSANT_ROOT` in the `.profile` file in your home directory:

```
VERSANT_ROOT=/usr/local/versant
export VERSANT_ROOT
```

Client-only, runtime installations need only the error message file under `VERSANT_ROOT`.

If you have static linkage for the executables, the rest of the Versant installation tree (folders) need not be present.

You can see the location of your Versant installation by typing:

```
oscp -p
```

## **VERSANT\_DBID\_NODE**

The parameter `VERSANT_DBID_NODE` sets the machine containing the `osc-dbid` database system file. This is a Mandatory Parameter.

This parameter is necessary if you want to create, remove, or list databases. There is no default location for the database system file.

If your machine is not connected to a network, you can specify this parameter with a null value to indicate that the `osc-dbid` file is on the local machine.

If your machine is connected to a network but will use a local database system file, you should supply the network name of your machine as known to your TCP/IP installation.

If your machine is connected to a network and will use a remote database system file, you should supply the network name of the remote machine as known to your TCP/IP installation.

You may set the parameter `VERSANT_DBID_NODE` in any of the following places:



---

**VERSANT\_DBID\_NODE**

In the environment variable `VERSANT_DBID_NODE`.

**user.cfg**

In a user configuration file, if you also set `VERSANT_CFG`.

`/etc/.oscxyyyzz`

In the machine configuration file, which is the default location.

**sysinfo**

In the `sysinfo` file in the `lib` directory.

For example, if you use C Shell and set `VERSANT_DBID_NODE` in your `.login` file in your home directory:

```
setenv VERSANT_DBID_NODE myserver
```

For example, if you use Bourne shell and set `VERSANT_DBID_NODE` in the `.profile` file in your home directory:

```
VERSANT_DBID_NODE = myserver
export VERSANT_DBID_NODE
```

**VERSANT\_DBID**

The parameter `VERSANT_DBID` sets the local or remote directory containing the `osc-dbid` database system file. This is a Mandatory Parameter.

This parameter is necessary if you want to create, remove, or list databases. There is no default location for the database system file.

Remote directories must be specified in terms understandable to the operating system on the remote machine. This means that for personal computers you should specify directories with `drive:\directory` syntax and for UNIX computers you should specify directories with `/directory` syntax.

You may set the parameter `VERSANT_DBID` in any of the following places:

**VERSANT\_DBID**

In the environment variable `VERSANT_DBID`.

**user.cfg**

In a user configuration file, if you also set `VERSANT_CFG`.

`/etc/.oscxyyyzz`

In the machine configuration file, which is the default location.

**sysinfo**

In the `sysinfo` file in the `lib` directory.

If the machine containing the `osc-dbid` file is running UNIX and the `osc-dbid` directory is `/usr/local/versant/db`, specify `VERSANT_DBID` as:

`VERSANT_DBID /usr/local/versant/db`

## VERSANT\_DB

The parameter `VERSANT_DB` sets the database root directory. This is a Mandatory Parameter.

This parameter is necessary if you want to access, create, or remove a database on your local machine. New databases are placed in a directory branching from the database root directory.

You may set the parameter `VERSANT_DB` in any of the following places:

**VERSANT\_DB**

In the environment variable `VERSANT_DB`.

**user.cfg**

In a user configuration file, if you also set `VERSANT_CFG`.

`/etc/.oscxyyyzz`

In the machine configuration file, which is the default location.

**sysinfo**

In the `sysinfo` file in the `lib` directory.

---

## VERSANT\_CFG

The name and location of the user configuration file containing specifications for software and database root directories. Only the following `VERSANT` environment variables can be set in the `VERSANT_CFG` file:

`VERSANT_ROOT`, `VERSANT_DB`, `VERSANT_DBID` and `VERSANT_DBID_NODE`.

All other environment variables need to be set outside `VERSANT_CFG`.

## UNIX Configuration Parameters - For Asserting Values

Following configuration parameters are needed only in a complex environment:

### VERSANT\_REL, VERSANT\_REL@*node*

#### Versant Release Number

The parameters `VERSANT_REL` and `VERSANT_REL@node` set the Versant release number. This parameter is normally not needed.

Versant allows you to assert a release number by setting the environment variable `VERSANT_REL` for a local machine or `VERSANT_REL@node` for a remote machine named `node`.

- You must assert a release number if your application was compiled with one Versant Release and you want to access a database of another Release.
- You may want to assert a release number if you have multiple Versant Releases on the same machine.

Asserting a release number causes Versant to use the configuration specified in the `.oscxyyyzz` machine configuration file that corresponds to the release number asserted.

Asserting a release number does not alter the normal order in which Versant searches for its configuration. If a configuration has been specified in environment variables or in a user configuration file, then asserting a release number will have no effect.

Except in the case when Release 6 applications are accessing Release 7 databases, the ability to assert release numbers is provided as a convenience. For example, when interacting with a remote machine, you may not know what values to assert for `VERSANT_ROOT@node`, `VERSANT_DB@node`, `VERSANT_DBID_NODE@node`, and `VERSANT_DBID@node` in order to completely

specify the remote configuration you want to use. In such a case, it is simpler just to assert a release number with `VERSANT_REL@node` and then let the remote machine determine the correct configuration to use.

### Background

A "complete" installation of Versant involves:

1. Copying Versant software to a machine, which is normally done for you by the installation program.
2. Setting at least the following configuration parameters: `VERSANT_ROOT`, which specifies the software root directory, `VERSANT_DBID_NODE`, which specifies the machine containing the `osc-dbid` file, `VERSANT_DBID`, which specifies the directory containing the `osc-dbid` file, and `VERSANT_DB`, which specifies the database root directory.

Unless you install manually, the installation program will set these parameters for you in the machine configuration file `/etc/.oscxyyzz` and system information file `/lib/sysinfo`. After installation, the combination of these two files completely defines the Versant configuration. The only additional environment change made during installation is to place the `/bin` directory on your path.

The ability to use a release number to assert a configuration depends upon the creation of the machine configuration file that contains the release number as part of its name.

3. Altering the TCP/IP configuration files to point to the server connection program in the Versant `/bin` directory for the release being installed.

After installing Versant, you must modify the TCP/IP configuration files manually. Normally you will update the `/etc/services` and `/etc/inetd.conf` files to point to the server connection program `ss.d` in the `../bin` directory of the new release. Doing this does not prevent you from using a previous installation of a Versant Release.

4. Converting any existing databases to be compatible with the installed release.

To be accessed by a particular version of Versant, a database must be converted to the same release as the server process that accesses it. After installing Versant, to use existing databases with the new release, you must manually convert existing databases with the `convertddb` utility provided in the new release.

If you want to first confirm the compatibility of various applications and databases, you may not want to perform this step right away, which creates the need to be able to assert a particular installation.

5. Deleting any previous installations of Versant software (but not the existing databases).

This must be done manually, although you should not delete the previous installation if you do not convert all existing databases to the new release.

---

Once you have installed the new version of Versant, when an application client process requests a database connection, the following happens:

1. The connection request is sent to the TCP/IP `inetd` daemon on the machine on which the database resides, which forwards the request to the server connection program specified in its configuration files.
2. The server connection program starts and looks for values for `VERSANT_ROOT`, `VERSANT_DB`, `VERSANT_DBID_NODE`, and `VERSANT_DBID`. The `ss` daemon will search in the following order:
  1. Environment variables
  2. User configuration file
  3. Machine configuration file
  4. System information file

Normally, if needed information is not found in environment variables or in a user configuration file, the server connection program will use the machine configuration file associated with its own release number.

However, if you have asserted a release number and needed information is not found in environment variables or in a user configuration file, the server connection program will use the machine configuration file associated with the release number that has been asserted.

3. The server connection program starts a database server process associated with the Versant configuration that it has found.

In general, client applications compiled with a lower release number can connect to database servers of a higher release number, but client applications cannot connect to a server of a lower release number. For example, a 6.0.5 client can connect to a 7.0 server, but a 7.0 client cannot connect with a 6.0.5 database.

### Setting `VERSANT_REL`

The general syntax for setting `VERSANT_REL` is:

```
VERSANT_REL xx.yy.zz
```

For `xx`, substitute the main release number; for `yy`, substitute the maintenance number; and for `zz` substitute the update number. Do not use leading zeroes.

For example, for Release 7.0.1:

```
VERSANT_REL 7.0.1
```

You must set the configuration parameter `VERSANT_REL` in the environment variable `VERSANT_REL`.

**Setting** `VERSANT_REL@node`

The general syntax for setting `VERSANT_REL@node` is:

`VERSANT_REL@node xx.yy.zz`

For `node`, substitute the name of the remote machine.

For `xx`, substitute the main release number; for `yy`, substitute the maintenance number; and for `zz` substitute the update number. Do not use leading zeroes.

For example, for Release 7.0.1 on a machine named `server`:

`VERSANT_REL@server 7.0.1`

You must set the configuration parameter `VERSANT_REL@node` in the environment variable `VERSANT_REL@node`.

**Environment setting on client machine**

The following shows how the `VERSANT_REL` setting on a server machine is affected by an environment variable setting on a client machine (`VERSANT_REL` or `VERSANT_REL@node`).

VERSANT_REL	VERSANT_REL@node	Value passed
Yes	No	None
Yes	Yes	VERSANT_REL@node
No	Yes	VERSANT_REL@node
No	No	none

**Example**

Suppose that a local machine and a remote machine named `alpha` each has more than one installation of Versant:

Client process uses Release 3.0.5

An application compiled with Release 3.0.5.

Local connection program uses Release 3.0.10

---

The local TCP/IP configuration files point to a server connection program in a Release 3.0.10 `bin` directory.

Desired local server processes use 3.0.8

You want to access local 3.0.8 databases.

Connection program on `alpha` uses 3.0.12

The TCP/IP configuration files on `alpha` point to a server connection program in a Release 3.0.12 `bin` directory.

Desired server processes on `alpha` use 3.0.10

You want to access 3.0.10 databases on `alpha`.

In this case, you must set the following:

```
VERSANT_REL 3.0.8
```

You must assert the local release to be 3.0.8 in order to start Release 3.0.8 server processes, because the server connection program is associated with Release 3.0.10.

```
VERSANT_REL@alpha 3.0.10
```

You must assert the release on `alpha` to be 3.0.10 in order to start Release 3.0.10 server processes, because the server connection program is associated with Release 3.0.12.

## **VERSANT\_ROOT@*node***

The parameter `VERSANT_ROOT@node` sets the Remote software root directory. This parameter is normally not needed.

This parameter is needed only if a remote machine named `node` has multiple versions of Versant software and you want to specify a particular software root directory.

Remote paths must be specified in syntax understood by the remote machine.

You may set the parameter `VERSANT_ROOT@node` in any of the following places:

**VERSANT\_ROOT@node**

In the environment variable `VERSANT_ROOT@node`.

**user.cfg**

In a user configuration file, if you also set `VERSANT_CFG`.

`/etc/.oscxyyyzz`

In the machine configuration file, which is the default location.

The client will be capable of sending environment of two server machines under the following criterion.

If the database network has two servers, imagine a server node 1 having the database(s) and another server node 2 having the `osc-dbid` file. In this case, if the client has environment set for both the machines `node 1` as well as `node 2` then the client will be able to send settings for both the machines.

The following shows how the `VERSANT_ROOT` setting on a server machine is affected by an environment variable setting on a client machine (`VERSANT_ROOT` or `VERSANT_ROOT@server`).

**Environment setting on client machine:**

(When the `osc-dbid` file and database(s) reside on the same machine).

VERSANT_ROOT	VERSANT_ROOT@node	Value passed
Yes	No	None
Yes	Yes	VERSANT_ROOT@node
No	Yes	VERSANT_ROOT@node
No	No	None

**Environment setting on client machine:**

(When the `osc-dbid` file is on `node2` and database(s) are on `node1`)

In this case utilities that need to access `osc-dbid` file will make two RPC calls. One from the client to the database server `node1`. Second from the database server `node1` to the machine having the `osc-dbid` file (i.e. `dbid` server) `node2`.



<b>VERSANT_ROOT (Client)</b>	<b>VERSANT_ROOT@node1 (Database server)</b>	<b>VERSANT_ROOT@node2 (dbid server)</b>	<b>Value passed (Client to node1)</b>	<b>Value passed (node1 to node2)</b>
Yes	Yes	Yes	VERSANT_ROOT@node1 VERSANT_ROOT@node2	VERSANT_ROOT@node2
Yes	Yes	No	VERSANT_ROOT@node1	None
Yes	No	Yes	None	None
Yes	No	No	None	None
No	Yes	Yes	VERSANT_ROOT@node1 VERSANT_ROOT@node2	VERSANT_ROOT@node2
No	Yes	No	VERSANT_ROOT@node1	None
No	No	Yes	None	None
No	No	No	None	None

## **NOTE:-**

1. If the `osc-dbid` file is on the client machine then node2=client and the above table will be still valid.
2. If node1 has settings for node2 then, the client need not maintain settings for node2. This way of configuration is also valid. However if the client has settings for node2 then the client settings will override settings of node2 on node1.

## **VERSANT\_DBID\_NODE@node**

The parameter `VERSANT_DBID_NODE@node` sets the Machine containing the database system file for a remote database. This parameter is normally not needed.

You need to specify this parameter only if there are multiple database systems and you want to override the specification for the machine containing the `osc-dbid` file that has been set on the remote machine named `node`.

If you supply the name of a remote database when you use the `create database`, `remove database`, or `list databases` utilities, Versant will look for the name of the machine containing the `osc-dbid` database system file in this parameter. If this parameter has not been set, Versant will look for the `osc-dbid` file in the location specified on the remote machine.

The general syntax for setting `VERSANT_DBID_NODE@node` is:

```
VERSANT_DBID_NODE@node osc-dbid_machine
```

Substitute for `node` the name of remote machine used with `dbname@node` syntax as an argument to the `createdb`, `removedb`, or `listdb` utilities. Substitute for `osc-dbid_machine` the name of the machine containing the `osc-dbid` file that you want to use.

You may set the parameter `VERSANT_DBID_NODE@node` in any of the following places:

### **VERSANT\_DBID\_NODE@node**

In the environment variable `VERSANT_DBID_NODE@node`.

### **user.cfg**

In a user configuration file, if you also set `VERSANT_CFG`.

```
/etc/.oscxyyz
```

In the machine configuration file, which is the default location.

### **sysinfo**

In the `sysinfo` file in the `lib` directory.

**For more information refer to “Database System Identifier File (osc-dbid)” on page 55, in “Chapter 2 - Database Creation”.**

### **VERSANT\_DBID@node**

The parameter `VERSANT_DBID@node` sets the directory containing the database system file for a remote database. This parameter is normally not needed.

As above, you need to specify this parameter only if there are multiple database systems and you want to override the specification for the machine containing the `osc-dbid` file that has been set on the remote machine named `node`.

---

If you supply the name of a remote database when you use the create database, remove database, or list databases utilities, Versant will look for the name of the directory containing the `osc-dbid` database system file in this parameter. If this parameter has not been set, Versant will look for the `osc-dbid` file in the location specified on the remote machine.

The general syntax for setting `VERSANT_DBID@node` is:

`VERSANT_DBID@node path`

Substitute for `node` the name of remote machine used with `dbname@node` syntax as an argument to the `createdb`, `removedb`, or `listdb` utilities. Substitute for `path` the directory containing the `osc-dbid` file in terms understandable to the operating system of the remote machine.

For example, if you want to create a database on a remote machine named `alpha` and specify an `osc-dbid` directory on a UNIX machine in `/usr/local/versant/db`:

`VERSANT_DBID@alpha /usr/local/versant/db`

You may set the parameter `VERSANT_DBID@node` in any of the following places:

`VERSANT_DBID@node`

In the environment variable `VERSANT_DBID@node`.

**user.cfg**

In a user configuration file, if you also set `VERSANT_CFG`.

`/etc/.oscxyzz`

In the machine configuration file, which is the default location.

**sysinfo**

In the `sysinfo` file in the `lib` directory.

See also “Database System Identifier File (`osc-dbid`)” on page 55 in “Chapter 2 - Database Creation”.

## **VERSANT\_DB@node**

The parameter `VERSANT_DB@node` sets the Database root directory for a remote database. This parameter is normally not needed.

You need to specify this parameter only if there are multiple database root directories on a remote machine and you want to override the specification that has been set on the remote machine named `node`.

If you specify a remote database in a command or method with the syntax `dbname@node`, Versant will look for a value set by this parameter. If this parameter has not been set, Versant will look for the database root directory in the location specified on the remote machine.

The general syntax for setting `VERSANT_DB@node` is:

```
VERSANT_DB@node path
```

For `node`, substitute the name of the remote machine. For `path`, substitute the location of the database root directory in terms understandable to the operating system of the remote machine.

For example, if you want to connect with a database on a UNIX machine named `alpha` whose database root directory is `/db`, set:

```
VERSANT_DB@alpha /db
```

If you want to connect with a database on an WINDOWS machine named `beta` whose database root directory is `c:\db`, set:

```
VERSANT_DB@beta c:\db
```

You may set the parameter `VERSANT_DB@node` in any of the following places:

### **VERSANT\_DB@node**

In the environment variable `VERSANT_DB@node`.

### **user.cfg**

In a user configuration file, if you also set `VERSANT_CFG`.

```
/etc/.oscxyyyzz
```

In the machine configuration file, which is the default location.

### **sysinfo**

In the `sysinfo` file in the `lib` directory.

---

## UNIX Error Logging and Debugging Parameters

### **versant\_fe\_errlog**

To turn error logging on for the application process, set the environment variable `versant_fe_errlog` to the name of the file where errors should be logged. By default, error logging is off.

On UNIX systems, create this environment variable with its name in all lower case.

In addition to error number and error message, entries in the error log file also include context information, such as operating system error number, file name, and so on. The exact context information varies from error to error.

This error logging mechanism should not be confused with the functionality represented by the file `LOGFILE`, which is used to report non-panic Versant system errors.

For UNIX systems, the general syntax to set this environment variable is similar to the following (depending upon your shell), where `filename` is the name of your error logging file:

```
setenv versant_fe_errlog filename
```

## UNIX Statistics Collection Parameters

### **VERSANT\_STAT\_FILE**

To collect performance and monitoring statistics and send them to a file, set the environment variable `VERSANT_STAT_FILE` to the name of a file and then run your application within its scope.

By default, all defined statistics will be collected. You may want to filter the statistics collected to reduce file size and improve application performance. To filter the statistics collected and set other collection parameters, you can use the `VERSANT_STAT_STATS`, `VERSANT_STAT_FUNCS`, `VERSANT_STAT_TIME`, `VERSANT_STAT_DBS`, and `VERSANT_STAT_FLUSH` environment variables. It is recommended to use this variable together with `VERSANT_STAT_STATS` variable to restrict the number of frontend statistics collected. This will avoid rapid growth of the file specified in `VERSANT_STAT_FILE` and get the necessary statistics without blocking the application performance.

If you set `VERSANT_STAT_FILE`, statistics will be collected during the time your application is in a database session.

The `VERSANT_STAT_FILE` environment variable is read each time a session starts, so if your application starts and stops a session, each new session will overwrite the file created by the previous session.

Within the file name, you can use the symbol `%p`, which will expand to your process identifier number. This can help create differing file names if you exit your application process between sessions, but it is not a guarantee because process `id` numbers can repeat.

Unlike other Versant environment variables, such as `VERSANT_ROOT`, the `VERSANT_STAT_FILE` environment variable is not read automatically when an application makes a connection to a database on a local or remote machine. Instead, `VERSANT_STAT_FILE` is a normal environment variable associated with a particular application process.

If `VERSANT_STAT_FILE` has been set, statistics are collected and written whenever a collection point is encountered by your application. A default collection point is encountered when your application enters or exits a Versant function. You can set additional collection points by inserting `o_autostatsenter()` and `o_autostatsexit()` function calls in your code.

Setting `VERSANT_STAT_FILE` has no effect on statistics collection performed by other mechanisms, such as a separate invocation of the `vstats` utility.

Setting `VERSANT_STAT_FILE` is the same as calling `o_autostatsbegin()` at the beginning of your session, except that it uses parameters found in `VERSANT_STAT_STATS`, `VERSANT_STAT_FUNCS`, `VERSANT_STAT_TIME`, `VERSANT_STAT_DBS`, and `VERSANT_STAT_FLUSH`.

Even when database replication has been turned on, statistics collection tools and mechanisms operate only on named databases and not on replica databases. To collect statistics for a replica database, you must apply statistics collection mechanisms specifically to the replica database. If the replica database goes down, statistics collection will stop and then restart when the database comes back up.

**For more information on Statistics collection and its usage, please refer to the chapter “Statistics Collection” in the *Versant Database Fundamentals Manual*.**

## VERSANT\_STAT\_STATS

If `VERSANT_STAT_FILE` has been set, by default all statistics are collected and sent to the named file whenever a collection point is encountered. To filter what application, session, connection, and database statistics are sent to a file, set the `VERSANT_STAT_STATS` environment variable to a space delimited list of the names of the statistics you want to collect.

When statistic names are specified in the environment variable `VERSANT_STAT_STATS`, they must be entered in lowercase without the leading `STAT_` prefix.

For example on UNIX platforms:

---

```
setenv VERSANT_STAT_STATS "se_net_bytes_read se_net_bytes_written"
```

Setting `VERSANT_STAT_STATS` has no effect on what statistics are collected by other mechanisms, such as a separate invocation of the `vstats` utility. However, you can call `o_collectstats()` and explicitly turn collection off for specific statistics, which will override the values set by `VERSANT_STAT_STATS`.

**For more information on Statistics collection and its usage, please refer to the chapter “Statistics Collection” in the *Versant Database Fundamentals Manual*.**

## VERSANT\_STAT\_FUNCS

If `VERSANT_STAT_FILE` has been set, by default all statistics are collected and sent to the named file whenever a collection point is encountered. To filter what function statistics are sent to the file, set the `VERSANT_STAT_FUNCS` environment variable to a space delimited list of the names of the function statistics you want to collect.

When function names are specified in the environment variable `VERSANT_STAT_FUNCS`, they must be entered in lowercase, with a leading `o_` prefix and without the leading `API_` prefix.

For example:

```
setenv VERSANT_STAT_FUNCS 'o_xact o_select'
```

Setting `VERSANT_STAT_FUNCS` has no effect on what statistics are collected by other mechanisms, such as a separate invocation of the `vstats` utility. However, you can call `o_collectstats()` and explicitly turn collection off for specific statistics, which will override the values set by `VERSANT_STAT_FUNCS`.

**For more information on Statistics collection and its usage, please refer to the chapter “Statistics Collection” in the *Versant Database Fundamentals Manual*.**

## VERSANT\_STAT\_TIME

If `VERSANT_STAT_FILE` has been set, by default statistics are collected and sent to the named file whenever a collection point is encountered. To set a collection interval, set `VERSANT_STAT_TIME` to a desired number of seconds. After `VERSANT_STAT_TIME` has been set, statistics will not be written unless the specified number of seconds have elapsed since the last time that statistics were written.

Setting a time interval will improve performance, but the trace-like information on which functions were called will not be recorded. It is not necessary to restrict time intervals to be integers.

For example:

```
setenv VERSANT_STAT_TIME 1.5
```

**For more information on Statistics collection and its usage, please refer to the chapter “Statistics Collection” in the *Versant Database Fundamentals Manual*.**

## VERSANT\_STAT\_DBS

If `VERSANT_STAT_FILE` has been set, by default statistics are collected for all databases to which an application makes a connection, including the session database. To limit statistics collection to specific databases, set this environment variable to a space delimited list of database names. For remote databases, use the syntax `dbname@node`.

For example:

```
setenv VERSANT_STAT_DBS 'db1@server1 db2@server2'
```

Statistics collection for a named database will begin when a connection is made to it and stop when a connection to it is closed.

To ensure that only application and session statistics are sent to file, set `VERSANT_STAT_DBS` to an empty string:

```
setenv VERSANT_STAT_DBS ''
```

**For more information on Statistics collection and its usage, please refer to the chapter “Statistics Collection” in the *Versant Database Fundamentals Manual*.**

## VERSANT\_STAT\_FLUSH

If `VERSANT_STAT_FLUSH` exists, each batch of statistics is sent to file as soon as it is generated. This eliminates a possible lag time caused by operating system file buffering, which is useful when you want to view the automatic collection file as it is being generated.

**For more information on Statistics collection and its usage, please refer to the chapter “Statistics Collection” in the *Versant Database Fundamentals Manual*.**



---

## UNIX User Authentication Parameter

### VERSANT\_AUTH

As a part of the process of customizing user authentication, on the application machine, set the environment variable `VERSANT_AUTH` to point to the application authentication program.

The general format for the environment variable is:

```
VERSANT_AUTH=program_path
```

**For more information on Authentication, please refer to the chapter “Controlling Access to Database” in the *Versant Database Fundamentals Manual*.**

## UNIX Communications Parameter

### Versant TCP/IP service name

This parameter provides support for multiple Versant releases.

The environment variable `VERSANT_SERVICE_NAME` allows you to have multiple Versant releases on the same machine and connect to servers from one or the other of these releases selectively from different clients. This is an advanced option and requires careful configuration. This is not something you normally need to do, and we recommend that you do it only with the guidance of Versant Technical Support.

In the default installation, the Versant connection services daemon listens on TCP/IP port 5019, which has the TCP/IP service name "oscssd" associated with it. In the default case, all clients try to connect to the "oscssd" service on the server machine, and the service name is mapped to a port number by looking up the TCP/IP configuration file `/etc/services` on the client machine (not the server machine).

By specifying the environment variable `VERSANT_SERVICE_NAME`, you can cause a client application to connect to a different service name and, therefore, a different port number. Correspondingly, on the server machine you can set up different Versant releases to listen on different port numbers and allow a client to connect to a Versant server process from the release of their choice.

### Example

For example, suppose that you have Versant Release 6.0.5 as well as 7.0.1 installed on the same server machine. The following procedure illustrates how to set up the server machine and the client machines for selective Versant release execution.

In this example, let's assume that the two releases have different `VERSANT_ROOT` directories. Suppose that Release 6.0.5 resides at `/usr/versant/6.0.5/sun4`, and Release 7.0.1 resides at `/apps/versant`.

In the following description, the sections on configuring TCP/IP services apply only to Unix machines. On Windows, a GUI utility named `networkservices.exe` is provided to configure Windows machines to run multiple Versant releases. This Windows utility is documented with other DBA utilities.

### Configuring a server machine

1. Select port numbers and service names.

First, select a port number for each release. For this example, let's choose port 5020 for Release 6.0.5 and port 5030 for Release 7.0.1.

Next, select service names to associate with these two ports. For this example, let's choose the service names `osc605` and `osc701`.

2. Get root privilege on your server machine.
3. Edit the `/etc/services` file.

Edit the file `/etc/services` and associate the service names with the chosen port numbers by adding the following two lines:

```
osc605  5020/tcp      # Versant 6.0.5 connection service
osc701  5030/tcp      # Versant 7.0.1 connection service
```

4. Edit the `/etc/inetd.conf` file.

Edit the file `/etc/inetd.conf` and associate each service name with the desired release by adding the following two lines:

#### For Solaris / HP-UX / AIX:

```
osc605 stream tcp nowait root /usr/versant/6.0.5/sun4/bin/ss.d
in.oscssd
osc701 stream tcp nowait root /apps/versant/bin/ss.d in.oscssd
```

#### For Linux:

```
osc605 stream tcp nowait root /usr/versant/6.0.5/sun4/bin/ss.d
osc701 stream tcp nowait root /apps/versant/bin/ss.d
```

You must provide the absolute path to the `ss.d` executable in the Versant `bin` directory for each release.

---

5. Refresh `inetd`.

Notify `inetd` that it should refresh its configuration, which we changed in the previous step:

```
ps -ef | grep inetd # Find the process id of inetd
kill -HUP <inetd-process-id> # Ask inetd to refresh
```

## Configuring a client machine

1. Edit the `services` file.

On each client machine, edit the `services` file and introduce the new service names and port numbers as in Steps 2 and 3 above. The service names and port numbers for Versant services must match each other exactly on the client and server machines.

## Usage at run-time

The following applies to both Unix and Windows platforms.

1. Set `VERSANT_SERVICE_NAME` on each client machine.

In an appropriate context on the 6.0.5 client machine, set the environment variable `VERSANT_SERVICE_NAME` to the value `"osc605"` before running a Release 6.0.5 application. This will automatically connect the application to the server database process for Versant Release 6.0.5.

In an appropriate context on the 7.0.1 client machine, set the environment variable `VERSANT_SERVICE_NAME` to the value `"osc701"` before running a Release 7.0.1 application. This will automatically connect the application to the server database process for Versant Release 7.0.1.

### For example:

#### On UNIX running C shell:

```
setenv VERSANT_SERVICE_NAME osc701
```

#### On UNIX running Bourne shell:

```
VERSANT_SERVICE_NAME=osc701; export
```

#### On Windows:

For a persistent change, go to Control Panel / System / Environment, enter `VERSANT_SERVICE_NAME` in the Variable entry box, enter the new service name in the Value entry box, click on Set and then click on either Apply or OK.

For a temporary change, you can set environment variables in a program. For example:

```
SetEnvironmentVariable("VERSANT_SERVICE_NAME", "osc701");  
putenv("VERSANT_SERVICE_NAME=osc701");
```

In the above, note that you need to execute both `SetEnvironmentVariable()`, in order to set the variable in the Windows process header, and `putenv()`, in order to set the variable in the C runtime environment.

## VERSANT\_SERVER\_PORTS

This feature allows users to access database servers that are protected by a firewall.

The `VERSANT_SERVER_PORTS` variable allows users to specify a list of port numbers/ranges. These port numbers are used to establish connections between the database server and the application processes or database utilities. These ports need to be opened through the firewall to make connections. The first available port in the list is used to make connection to the server. If no unused port is found, an error `NET_NO_UNUSED_PORT` is returned.

`VERSANT_SERVER_PORTS` is a comma separated list of port numbers or port ranges.

The following are the rules that apply for values that can be assigned to `VERSANT_SERVER_PORTS`:

- The value can contain digits between 0 to 9, comma(',') and hyphen('-')
- Any other character will be treated as an invalid value.
- The hyphen has to be preceded and followed by a number, else the value will be treated as invalid.
- If a range is specified, the start value has to be less than or equal to the end value.  
For E.g. 40010-4005 will be treated as an invalid value.
- In any of the configuration files, the first space or tab is considered as the end of value specification. Thus, if the value contains space/tab then only the part prior to space/tab will be considered.

For E.g.: `VERSANT_SERVER_PORTS=40000,40005-40010,40026-40050,<tab>50000-50010`

In this case only "40000,40005-40010,40026-40050," will be considered, "50000-50010" will be ignored.

In case of invalid value for `VERSANT_SERVER_PORTS`, error `NET_EINVALIDPORTS` is thrown.

User may set the parameter `VERSANT_SERVER_PORTS` in any of the following places:

- In the user environment variable `VERSANT_SERVER_PORTS`.
- In a user configuration file set through `VERSANT_CFG`
- In the machine configuration file `-/etc/.oscxyyzz`
- In the `VERSANT` system information file "sysinfo" in the lib directory.

The value for `VERSANT_SERVER_PORTS` will be searched in the following priority order:

1. Environment variables
2. User configuration file
3. Machine configuration file
4. System information file

If `VERSANT_SERVER_PORTS` is not defined, `VERSANT` will allow the operating system to assign any arbitrary port for communication with the application.

This configuration parameter should be used instead of `obe_port_begin` and `obe_port_end` parameters in the `profile.be` file of the database. The parameters `obe_port_begin` and `obe_port_end` are no longer supported.

### **Example for setting `VERSANT_SERVER_PORTS`:**

#### **On running C Shell**

To set `VERSANT_SERVER_PORTS` in the `.login` file in the home directory:

```
setenv VERSANT_SERVER_PORTS 40000,40005-40010,40026-40050
```

#### **On running Bourne Shell**

To set `VERSANT_SERVER_PORTS` in the `.profile` file in the home directory:

```
VERSANT_SERVER_PORTS = 40000,40005-40010,40026-40050
export VERSANT_SERVER_PORTS
```

## **UNIX Examples**

### **UNIX example using normal parameters**

Suppose you are running on a UNIX machine named `myMachine`, you are accessing an WINDOWS machine named `alpha` and a UNIX machine named `beta`, and the `osc-dbid` database system file is on `beta` in `/dbsys`.

To specify information about your machine, you must set:

```
VERSANT_ROOT      /usr/local/versant
PATH              <existing PATH>:/usr/local/versant/
7.0.1/sun4/bin
```

If you want to create databases on your local machine, you must set:

```
VERSANT_DBID_NODE  beta
VERSANT_DBID       /dbsys
VERSANT_DB         /usr/local/versant/db
```

After setting all of these variables, if you invoke `createdb` as:

```
createdb myDatabase
```

Then Versant will look at the following variables:

### **PATH**

To find the `createdb` utility.

### **VERSANT\_DBID\_NODE**

To find the machine containing the `osc-dbid` file.

### **VERSANT\_DBID**

To find the directory containing the `osc-dbid` file.

### **VERSANT\_DB**

To find the database root directory.

After setting all of these variables, invoke `createdb` as:

```
createdb myDatabase@alpha
```

After making the connection, Versant will pass the `createdb` command to `alpha`. Machine `alpha` will then look to its own configuration parameters to decide where to find the `osc-dbid` file and where to place the new database.

## UNIX example using advanced parameters

Suppose your UNIX machine is named `myMachine` and you are accessing a WINDOWS machine named `alpha` and a UNIX machine named `beta`, and the `osc-dbid` database system file is on `beta` in `/dbsys`.

If you have multiple releases of Versant on your local machine or databases created with multiple releases and want to assert the release number, you may optionally set:

```
VERSANT_REL          7.0.1
```

If you want to assert the location of the `osc-dbid` database system file to machines `alpha` and `beta`, you may optionally set:

```
VERSANT_DBID_NODE@alpha  beta
VERSANT_DBID@alpha       /dbsys
VERSANT_DBID_NODE@beta   beta
VERSANT_DBID@beta        /dbsys
```

If you want to assert the location of database root directories to machines `alpha` and `beta`, you may optionally set:

```
VERSANT_DB@alpha        d:\versant\db
VERSANT_DB@beta          /versant/db
```

If you have multiple releases of Versant on `alpha` and `beta` and databases created with multiple releases, to assert the release number and root directories, you may optionally set:

```
VERSANT_REL@alpha       6.0.5
VERSANT_ROOT@alpha       d:\versant
VERSANT_ROOT@beta        /usr/local/versant
VERSANT_REL@beta         7.0.1
```

After setting all of the above parameters, if you invoke `createdb` as:

```
createdb myDatabase@alpha
```

Then Versant will look at the following parameters:

### **VERSANT\_REL@alpha**

To assert the release number on `alpha`.

### **VERSANT\_ROOT@alpha**

To assert the software root directory on `alpha`.

**VERSANT\_DBID\_NODE@alpha**

To assert the `osc-dbid` machine.

**VERSANT\_DBID@alpha**

To assert the `osc-dbid` directory.

**VERSANT\_DB@alpha**

To assert the database root directory.

## UNIX Database Server System Parameter

**VERSANT\_BE\_STACK\_SIZE**

Stack size of the Versant Object Database server thread. Optional.

This configuration parameter controls the stack size of the threads that are created by the Versant database server process (`obe`). It is expressed in terms of kilo bytes (KB) and the default size is 256. You can specify a value between 256KB and 1024 KB. If a value less than 256KB is specified then 256KB will be used and if a value greater than 1024KB is specified then 1024KB will be used.

You may set the parameter `VERSANT_BE_STACK_SIZE` in:

`/etc/.oscxyyyzz`

In the machine configuration file, which is the default location.

For example, if you use C Shell and set `VERSANT_BE_STACK_SIZE` to 512

```
setenv VERSANT_BE_STACK_SIZE 512
```

For example, if you use Bourne shell and set `VERSANT_BE_STACK_SIZE` to 512

```
VERSANT_ROOT=512
export VERSANT_ROOT
```



---

## CONFIGURATION PARAMETERS FOR WINDOWS

The following explains configuration parameters for Windows operating systems.

Since the personal computer operating systems differ on how they set and use parameters, please see the Release notes for information on where and how to set configuration parameters. This section just discusses the parameters that can be set and what they do.

### Windows Configuration Parameter Considerations

#### Passing parameters to remote machines

To specify a command or configuration parameter on a remote machine, you can use `@node` syntax. For example, to create a database on a remote machine named `server`, you would use the following `@node` syntax:

```
createdb db1@server
```

Or, to look for an `osc-dbid` file on a remote machine:

```
VERSANT_DBID_NODE@server
```

If you set a configuration parameter without `@node` syntax, that parameter applies only to the local machine. If you use `@node` syntax, the value is sent to the remote machine for its use.

Configuration parameters explicitly passed to a remote machine override those already set on the remote machine. You should pass parameters only if you want to override the values set on the remote machine.

For example, suppose an Windows machine named `server` has the following environment parameter set:

```
VERSANT_DB \versant\db
```

To create a database on `server` in a different directory, you could set the following:

```
VERSANT_DB@server \db
```

Then, if you invoke the create database utility as `createdb@server`, the value of your environment parameter `VERSANT_DB@server` will be passed to the machine named `server` and the database will be created in a directory branching from `\db` rather than `\versant\db`.

You can use `@node` syntax for any parameter and set any parameter for remote use in either a configuration file or in an environment variable.

If you doubly define a parameter for the local machine by defining it both with and without `@node` syntax, Versant will look first for a value in the parameter without `@node` syntax and then look for it with `@node` syntax. The first value found would be the value used.

For example, if you are on machine `alpha` and have defined both `VERSANT_DB` and `VERSANT_DB@alpha`, then Versant will use the value in `VERSANT_DB`. However, if you have not defined `VERSANT_DB`, then Versant will use the value in `VERSANT_DB@alpha`.

In any case, you must specify parameters to be passed in the syntax appropriate to the remote machine. For example:

```
VERSANT_DB@unix_server /usr/local/versant/db
VERSANT_DB@windows_server \\versant\db
```

## Windows Configuration Parameters - Mandatory

The following configuration parameters are necessary for Versant to run in a standard environment.

### PATH

The parameter `Path` sets the Executables directory. This is always needed.

After installation, append the location of Versant executables directory `bin` to your `PATH` environment variable. This will enable Versant to find the executables regardless of your current directory.

## Windows Configuration Parameters - To be set in Special Cases

The following configuration parameters are necessary for Versant to run in a standard environment.

You need to explicitly set the windows parameters only in the following situations:

- If you did not answer all prompts during installation,
- If you made a mistake during installation,
- If you have moved your software, database directory, or database system identifier file.

---

When you install Versant, all configuration parameters needed by Versant are set and stored.

**For more information, on Versant file structures, please refer to Chapter 4 “Directories and Files” on page 281.**

## VERSANT\_ROOT

The parameter `VERSANT_ROOT` sets the software root directory. This is a Mandatory Parameter.

This parameter enables Versant to find its own files, because they are placed in predetermined places in directories built from the software root directory.

You must specify the location of the software root directory by setting the environment variable `VERSANT_ROOT`.

The general syntax is:

```
set VERSANT_ROOT=drive:directory
```

For example:

```
set VERSANT_ROOT=c:\versant
```

You cannot set `VERSANT_ROOT` in the `sysinfo` file.

Client-only, runtime installations need only the error message file under `VERSANT_ROOT`. If you have static linkage for the executables, the rest of the Versant installation tree does not have to be present.

You can see the location of your Versant installation by typing:

```
oscp -p
```

## VERSANT\_DBID\_NODE

The parameter `VERSANT_DBID_NODE` sets the machine containing the `osc-dbid` database system file. This parameter is usually needed.

This parameter is necessary if you want to create, remove, or list databases. There is no default location for the database system file.

If your machine is not connected to a network, you can skip specifying this parameter, which will indicate that the `osc-dbid` file is on the local machine.

If your machine is connected to a network but will use a local database system file, you should supply the network name of your machine as known to your TCP/IP installation.

If your machine is connected to a network and will use a remote database system file, you should supply the network name of the remote machine as known to your TCP/IP installation.

You can set `VERSANT_DBID_NODE` as an environment variable or with a line in the `sysinfo` file. The general syntax is:

```
VERSANT_DBID_NODE osc-dbid_machine
```

### VERSANT\_DBID

The parameter `VERSANT_DBID` sets the local or remote directory containing the `osc-dbid` database system file. This parameter is usually needed.

This parameter is necessary if you want to create, remove, or list databases. There is no default location for the database system file.

Remote directories must be specified in terms understandable to the operating system on the remote machine. This means that for personal computers you should specify directories with `drive:\directory` syntax and for UNIX computers you should specify directories with `/directory` syntax.

You can set `VERSANT_DBID_NODE` as an environment variable or with a line in the `sysinfo` file. The general syntax is:

```
VERSANT_DBID path
```

For `path`, substitute the location of the software root directory in terms understandable to the operating system of the machine containing the `osc-dbid` file.

For example, if the machine containing the `osc-dbid` file is a personal computer and the `osc-dbid` directory on that machine is `c:\versant\db`:

```
VERSANT_DBID c:\versant\db
```

If the machine containing the `osc-dbid` file is running UNIX and the `osc-dbid` directory is `/usr/local/versant/db`, specify `VERSANT_DBID` as:

```
VERSANT_DBID /usr/local/versant/db
```

### VERSANT\_DB

The parameter `VERSANT_DB` sets the database root directory. This parameter is usually needed.

This parameter is necessary if you want to create or remove a database on your local machine. New databases are placed in a directory branching from the database root directory.

---

You can set `VERSANT_DB` as an environment variable or with a line in the `sysinfo` file. The general syntax is:

```
VERSANT_DB path
```

For `path`, substitute the location of the database root directory.

For example:

```
VERSANT_DB c:\versant\db
```

## VERSANT\_USER

The parameter `VERSANT_USER` sets the Network username. This parameter is Necessary for remote connections.

On a personal computer, you must set this parameter if you want to access, create, or remove a database on a remote machine.

If you are using Versant on a standalone basis, you do not have to specify a username.

You can set `VERSANT_USER` with a line in the `sysinfo` file or by setting an environment variable. The general syntax is:

```
VERSANT_USER username
```

Substitute for `username` the login name to be used when you make a database connection.

For example, for the login name `silas`:

```
VERSANT_USER silas
```

## VERSANT\_HOST\_NAME

If the `VERSANT_HOST_NAME` is set, Versant uses this value instead of the actual hostname for the local machine. The `VERSANT_HOST_NAME` can be used for clustered environments and for resolving Media Sensing related issues.

Media Sensing is a new feature on Windows 2000 and XP that disables the network card when the PC is not connected to the network. Since Versant internally relies on TCP/IP and the fact that the IP address does not change during the runtime of the application and database, this causes the application to get Network related Versant error message like E3004 or E3005.

In order to solve this problem, customer can use the Versant environment variable `VERSANT_HOST_NAME`. It should be set to localhost i.e. 127.0.0.1 on the PC before the Versant cli-

ent and server is started and after that even if the network cable is unplugged, application will not get network exceptions.

`VERSANT_HOST_NAME` can be used to solve the network-related problems, in case the application is using local databases.

## Windows Configuration Parameters - Optional

On personal computers, Versant also uses the following non-Versant environment variables.

### TZ

Your local time zone is needed by the C++/Versant `VDate` and `VTime` classes.

For example, for Pacific time, set `TZ` to equal `PST8PDT`.

### HOSTNAME

The TCP/IP name of your local machine, which is needed by TCP/IP network software.

### ETC

The TCP/IP `etc` directory, which is needed by TCP/IP network software.

The Versant installation program will optionally update your TCP/IP services file as appropriate for your operating environment. TCP/IP services are required for other clients to remotely access databases local to your personal computer environment or if you link your application to the Versant two process system library (the usual choice).

If you elect to update the TCP/IP services file yourself, want to examine Versant's changes, or need to change the Versant installation, the following changes are needed for each of the personal computer operating systems supported by Versant.

### WINDOWS

In Windows, the `\tcpip\etc\services` file should define Versant service as: `oscssd 5019/tcp`

In Windows, the `\\%SYSTEMROOT%\system32\drivers\etc\services` file should define Versant service as: `oscssd 5019/tcp`

---

## Windows Configuration Parameters - For Asserting Values

Following configuration parameters are needed only in a complex environment.

### **VERSANT\_REL, VERSANT\_REL@node**

The parameters `VERSANT_REL` and `VERSANT_REL@node` set the Versant release number. This parameter is normally not needed.

Versant allows you to assert a release number by setting the environment variable `VERSANT_REL` for a local machine or `VERSANT_REL@node` for a remote machine named `node`.

- You must assert a release number if your application was compiled with one Versant Release and you want to access a database of another Release.
- You may want to assert a release number if you have multiple Versant Releases on the same machine.

Asserting a release number causes Versant to use the configuration specified in `.oscxyyz` machine configuration file that corresponds to the release number asserted.

Asserting a release number does not alter the normal order in which Versant searches for its configuration. If a configuration has been specified in environment variables or in a user configuration file, then asserting a release number will have no effect.

Except in the case when Release 6 applications are accessing Release 7 databases, the ability to assert release numbers is provided as a convenience. For example, when interacting with a remote machine, you may not know what values to assert for `VERSANT_ROOT@node`, `VERSANT_DB@node`, `VERSANT_DBID_NODE@node`, and `VERSANT_DBID@node` in order to completely specify the remote configuration you want to use. In such a case, it is simpler just to assert a release number with `VERSANT_REL@node` and then let the remote machine determine the correct configuration to use.

**See also “VERSANT\_REL, VERSANT\_REL@node” on page 329, in the Section “Configuration Parameters for UNIX”.**

### **VERSANT\_ROOT@node**

The parameter `VERSANT_ROOT@node` sets the Remote software root directory. This parameter is normally not needed.

This parameter is needed only if a remote machine named `node` has multiple versions of Versant software and you want to specify a particular software root directory.

Remote paths must be specified in syntax understood by the remote machine.

You may set the parameter `VERSANT_ROOT@node` in an environment variable or in the `sysinfo` file.

The following shows how the `VERSANT_ROOT` setting on a server machine is affected by an environment variable setting on a client machine (`VERSANT_ROOT` or `VERSANT_ROOT@server`).

### Environment setting on client machine:

<b>VERSANT_ROOT</b>	<b>VERSANT_ROOT@node</b>	<b>Value passed</b>
Yes	No	None
Yes	Yes	<code>VERSANT_ROOT@node</code>
No	Yes	<code>VERSANT_ROOT@node</code>
No	No	None

### **VERSANT\_DBID\_NODE@*node***

The parameter `VERSANT_DBID_NODE@node` sets the Machine containing the database system file for a remote database. This parameter is normally not needed.

You need to specify this parameter only if there are multiple database systems and you want to override the specification for the machine containing `osc-dbid` file that has been set on the remote machine named `node`.

If you supply the name of a remote database when you use the create database, remove database, or list databases utilities, Versant will look for the name of the machine containing the `osc-dbid` database system file in this parameter. If this parameter has not been set, Versant will look for the `osc-dbid` file in the location specified on the remote machine.

You can set `VERSANT_DBID_NODE@node` as an environment variable or with a line in the `sysinfo` file. The general syntax is:

```
VERSANT_DBID_NODE@node osc-dbid_machine
```

Substitute for `node` the name of remote machine used with `dbname@node` syntax as an argument to the `createdb`, `removedb`, or `listdb` utilities. Substitute for `osc-dbid_machine` the name of the machine containing the `osc-dbid` file that you want to use.



---

For more information refer to “Database System Identifier File (osc-dbid)” on page 55, in “Chapter 2 - Database Creation”.

### **VERSANT\_DBID@node**

The parameter `VERSANT_DBID@node` sets the directory containing the database system file for a remote database. This parameter is normally not needed.

You need to specify this parameter only if there are multiple database systems and you want to override the specification for the machine containing the `osc-dbid` file that has been set on the remote machine named `node`.

If you supply the name of a remote database when you use the `create database`, `remove database`, or `list databases` utilities, Versant will look for the name of the directory containing the `osc-dbid` database system file in this parameter. If this parameter has not been set, Versant will look for the `osc-dbid` file in the location specified on the remote machine.

You can set `VERSANT_DBID@node` as an environment variable or with a line in a configuration file. The general syntax is:

```
VERSANT_DBID@node path
```

Substitute for `node` the name of remote machine used with `dbname@node` syntax as an argument to the `createdb`, `removedb`, or `listdb` utilities. Substitute for `path` the directory containing the `osc-dbid` file in terms understandable to the operating system of the remote machine.

For example, if you want to create a database on a remote machine named `alpha` and specify an `osc-dbid` directory on a UNIX machine in `/usr/local/versant/db`:

```
VERSANT_DBID@alpha /usr/local/versant/db
```

For more information refer to “Database System Identifier File (osc-dbid)” on page 55, in “Chapter 2 - Database Creation”.

### **VERSANT\_DB@node**

The parameter `VERSANT_DB@node` sets the Database root directory for a remote database. This parameter is normally not needed.

You need to specify this parameter only if there are multiple database root directories on a remote machine and you want to override the specification that has been set on the remote machine named `node`.

If you specify a remote database in a command or method with the syntax `dbname@node`, Versant will look for a value set by this parameter. If this parameter has not been set, Versant will look for the database root directory in the location specified on the remote machine.

You can set `VERSANT_DB@node` as an environment variable or with a line in the `sysinfo` file. The general syntax is:

```
VERSANT_DB@node path
```

For `node`, substitute the name of remote machine. For `path`, substitute the location of database root directory in terms understandable to the operating system of the remote machine.

For example, if you want to connect with a database on a UNIX machine named `alpha` whose database root directory is `/db`, set:

```
VERSANT_DB@alpha /db
```

If you want to connect with a database on a `WINDOWS` machine named `beta` whose database root directory is `c:\db`, set:

```
VERSANT_DB@beta c:\db
```

## Windows Error Logging and Debugging Parameters

### Versant\_fe\_errlog

To turn error logging on for the application process, set the environment variable `versant_fe_errlog` to the name of the file where errors should be logged. By default, error logging is off.

In addition to error number and error message, entries in the error log file also include context information, such as operating system error number, file name and so on. The exact context information varies from error to error.

This error logging mechanism should not be confused with the functionality represented by the file `LOGFILE`, which is used to report non-panic Versant system errors.

---

## Windows Statistics Collection Parameters

### VERSANT\_STAT\_FILE

To collect performance and monitoring statistics and send them to a file, set the environment variable `VERSANT_STAT_FILE` to the name of a file and then run your application within its scope.

By default, all defined statistics will be collected. You may want to filter the statistics collected to reduce file size and improve application performance. To filter the statistics collected and set other collection parameters, you can use the `VERSANT_STAT_STATS`, `VERSANT_STAT_FUNCS`, `VERSANT_STAT_TIME`, `VERSANT_STAT_DBS`, and `VERSANT_STAT_FLUSH` environment variables. It is recommended to use this variable together with `VERSANT_STAT_STATS` variable to restrict the number of frontend statistics collected. This will avoid rapid growth of the file specified in `VERSANT_STAT_FILE` and get the necessary statistics without blocking the application performance.

If you set `VERSANT_STAT_FILE`, statistics will be collected during the time your application is in a database session.

The `VERSANT_STAT_FILE` environment variable is read each time a session starts, so if your application starts and stops a session, each new session will overwrite the file created by the previous session.

Within the file name, you can use the symbol `%p`, which will expand to your process identifier number. This can help create differing file names if you exit your application process between sessions, but it is not a guarantee because process id numbers can repeat.

Unlike other Versant environment variables, such as `VERSANT_ROOT`, the `VERSANT_STAT_FILE` environment variable is not read automatically when an application makes a connection to a database on a local or remote machine. Instead, `VERSANT_STAT_FILE` is a normal environment variable associated with a particular application process.

If `VERSANT_STAT_FILE` has been set, statistics are collected and written whenever a collection point is encountered by your application. A default collection point is encountered when your application enters or exits a Versant function. You can set additional collection points by inserting `o_autostatsenter()` and `o_autostatsexit()` function calls in your code.

Setting `VERSANT_STAT_FILE` has no effect on statistics collection performed by other mechanisms, such as a separate invocation of the `vstats` utility.

Setting `VERSANT_STAT_FILE` is the same as calling `o_autostatsbegin()` at the beginning of your session, except that it uses parameters found in `VERSANT_STAT_STATS`, `VERSANT_STAT_FUNCS`, `VERSANT_STAT_TIME`, `VERSANT_STAT_DBS`, and `VERSANT_STAT_FLUSH`.

Even when database replication has been turned on, statistics collection tools and mechanisms operate only on named databases and not on replica databases. To collect statistics for a replica database, you must apply statistics collection mechanisms specifically to the replica database. If the replica database goes down, statistics collection will stop and then restart when the database comes back up.

**For more information on Statistics collection and its usage, please refer to the chapter “Statistics collection” in the *Versant Database Fundamentals Manual*.**

## VERSANT\_STAT\_STATS

If `VERSANT_STAT_FILE` has been set, by default all statistics are collected and sent to the named file whenever a collection point is encountered. To filter what application, session, connection, and database statistics are sent to a file, set the `VERSANT_STAT_STATS` environment variable to a space delimited list of the names of the statistics you want to collect.

When statistic names are specified in the environment variable `VERSANT_STAT_STATS`, they must be entered in lowercase without the leading `STAT_` prefix.

For example on Windows platform:

```
set VERSANT_STAT_STATS=se_net_bytes_read se_net_bytes_written
```

Setting `VERSANT_STAT_STATS` has no effect on what statistics are collected by other mechanisms, such as a separate invocation of the `vstats` utility. However, you can call `o_collectstats()` and explicitly turn collection off for specific statistics, which will override the values set by `VERSANT_STAT_STATS`.

**For more information on Statistics collection and its usage, please refer to the chapter “Statistics collection” in the *Versant Database Fundamentals Manual*.**

## VERSANT\_STAT\_FUNCS

If `VERSANT_STAT_FILE` has been set, by default all statistics are collected and sent to the named file whenever a collection point is encountered. To filter what function statistics are sent to the file, set the `VERSANT_STAT_FUNCS` environment variable to a space delimited list of the names of the function statistics you want to collect.

When function names are specified in the environment variable `VERSANT_STAT_FUNCS`, they must be entered in lowercase, with a leading `o_` prefix and without the leading `CAPI_` prefix.

For example:

---

```
set VERSANT_STAT_FUNCS=o_xact o_select
```

Setting `VERSANT_STAT_FUNCS` has no effect on what statistics are collected by other mechanisms, such as a separate invocation of the `vstats` utility. However, you can call `o_collectstats()` and explicitly turn collection off for specific statistics, which will override the values set by `VERSANT_STAT_FUNCS`.

**For more information on Statistics collection and its usage, please refer to the chapter “Statistics collection” in the *Versant Database Fundamentals Manual*.**

## VERSANT\_STAT\_TIME

If `VERSANT_STAT_FILE` has been set, by default statistics are collected and sent to the named file whenever a collection point is encountered. To set a collection interval, set `VERSANT_STAT_TIME` to a desired number of seconds. After `VERSANT_STAT_TIME` has been set, statistics will not be written unless the specified number of seconds have elapsed since the last time that statistics were written.

Setting a time interval will improve performance, but the trace-like information on which functions were called will not be recorded. It is not necessary to restrict time intervals to be integers.

For example:

```
set VERSANT_STAT_TIME=1.5
```

**For more information on Statistics collection and its usage, please refer to the chapter “Statistics collection” in the *Versant Database Fundamentals Manual*.**

## VERSANT\_STAT\_DBS

If `VERSANT_STAT_FILE` has been set, by default statistics are collected for all databases to which an application makes a connection, including the session database. To limit statistics collection to specific databases, set this environment variable to a space delimited list of database names. For remote databases, use the syntax `dbname@node`.

For example:

```
set VERSANT_STAT_DBS=db1@server1 db2@server2
```

Statistics collection for a named database will begin when a connection is made to it and stop when a connection to it is closed.

To ensure that only application and session statistics are sent to file, set `VERSANT_STAT_DBS` to an empty string:

```
set VERSANT_STAT_DBS=
```

For more information on Statistics collection and its usage, please refer to the chapter “Statistics collection” in the *Versant Database Fundamentals Manual*.

## VERSANT\_STAT\_FLUSH

If `VERSANT_STAT_FLUSH` exists, each batch of statistics is sent to file as soon as it is generated. This eliminates a possible lag time caused by operating system file buffering, which is useful when you want to view the automatic collection file as it is being generated.

For more information on Statistics collection and its usage, please refer to the chapter “Statistics collection” in the *Versant Database Fundamentals Manual*.

## Windows User Authentication Parameter

### VERSANT\_AUTH

As a part of the process of customizing user authentication, on the application machine, set the environment variable `VERSANT_AUTH` to point to the application authentication program.

The general format for the environment variable is:

```
VERSANT_AUTH=program_path
```

## Windows Parameters

### VERSANT\_SERVICE\_NAME

#### Versant TCP/IP service name

This parameter provides support for multiple Versant releases.

The environment variable `VERSANT_SERVICE_NAME` allows you to have multiple Versant releases on the same machine and connect to servers from one or the other of these releases selectively from different clients. This is an advanced option and requires careful configuration. This is not something you normally need to do, and we recommend that you do it only with the guidance of Versant Technical Support.

---

In the default installation, the Versant connection services daemon listens on TCP/IP port 5019, which has the TCP/IP service name "osc`ssd`" associated with it. In the default case, all clients try to connect to the "osc`ssd`" service on the server machine, and the service name is mapped to a port number by looking up the TCP/IP configuration file `/etc/services` on the client machine (not the server machine).

By specifying the environment variable `VERSANT_SERVICE_NAME`, you can cause a client application to connect to a different service name and, therefore, a different port number. Correspondingly, on the server machine you can set up different Versant releases to listen on different port numbers and allow a client to connect to a Versant server process from the release of their choice.

### Example

For example, suppose that you have Versant Release 6.0.5 as well as 7.0.1 installed on the same server machine. The following procedure illustrates how to set up the server machine and the client machines for selective Versant release execution.

In this example, let's assume that the two releases have different `VERSANT_ROOT` directories. Suppose that Release 6.0.5 resides at `/usr/versant/6.0.5/sun4`, and Release 7.0.1 resides at `/apps/versant`.

In the following description, the sections on configuring TCP/IP services apply only to Unix machines. On Windows, a GUI utility named `networkservices.exe` is provided to configure Windows machines to run multiple Versant releases. This Windows utility is documented with the other DBA utilities.

### Configuring a server machine

1. Select port numbers and service names.

First, select a port number for each release to listen on. For this example, let's choose port 5020 for Release 6.0.5 and port 5030 for Release 7.0.1.

Next, select service names to associate with these two ports. For this example, let's choose the service names `osc605` and `osc701`.

2. Get root privilege on your server machine.
3. Edit the `/etc/services` file.

Edit the file `/etc/services` and associate the service names with the chosen port numbers by adding the following two lines:

```
osc605  5020/tcp      # Versant 6.0.5 connection service
osc701  5030/tcp      # Versant 7.0.1 connection service
```

4. Edit the `/etc/inetd.conf` file.

Edit the file `/etc/inetd.conf` and associate each service name with the desired release by adding the following two lines:

**For Solaris / HP-UX / AIX:**

```
osc605 stream tcp nowait root /usr/versant/6.0.5/sun4/bin/ss.d  
in.oscssd
```

```
osc701 stream tcp nowait root /apps/versant/bin/ss.d in.oscssd
```

**For Linux:**

```
osc605 stream tcp nowait root /usr/versant/6.0.5/sun4/bin/ss.d
```

```
osc701 stream tcp nowait root /apps/versant/bin/ss.d
```

You must provide the absolute path to the `ss.d` executable in the Versant `bin` directory for each release.

**5. Refresh inetd.**

Notify `inetd` that it should refresh its configuration, which we changed in the previous step:

```
ps -ef | grep inetd # Find the process id of inetd  
kill -HUP <inetd-process-id> # Ask inetd to refresh
```

### Configuring a client machine

**1. Edit the services file.**

On each client machine, edit the `services` file and introduce the new service names and port numbers as in Steps 2 and 3 above. The service names and port numbers for Versant services must match each other exactly on the client and server machines.

### Usage at run-time

The following applies to both Unix and Windows platforms.

**1. Set `VERSANT_SERVICE_NAME` on each client machine.**

In an appropriate context on the 6.0.5 client machine, set the environment variable `VERSANT_SERVICE_NAME` to the value `"osc605"` before running a Release 6.0.5 application. This will automatically connect the application to the server database process for Versant Release 6.0.5.

In an appropriate context on the 7.0.1 client machine, set the environment variable `VERSANT_SERVICE_NAME` to the value `"osc701"` before running a Release 7.0.1 application. This will automatically connect the application to the server database process for Versant Release 7.0.1.

**For example:**



---

**On UNIX running C shell:**

```
setenv VERSANT_SERVICE_NAME osc701
```

**On UNIX running Bourne shell:**

```
VERSANT_SERVICE_NAME=osc701; export
```

**On Windows:**

For a persistent change, go to Control Panel / System / Environment, enter `VERSANT_SERVICE_NAME` in the Variable entry box, enter the new service name in the Value entry box, click on Set and then click on either Apply or OK.

For a temporary change, you can set environment variables in a program. For example:

```
SetEnvironmentVariable("VERSANT_SERVICE_NAME", "osc701");  
putenv("VERSANT_SERVICE_NAME=osc701");
```

In the above, note that you need to execute both `SetEnvironmentVariable()`, in order to set the variable in the Windows process header, and `putenv()`, in order to set the variable in the C runtime environment.

## VERSANT\_SERVER\_PORTS

This feature allows users to access database servers that are protected by a firewall.

The `VERSANT_SERVER_PORTS` variable allows users to specify a list of port numbers/ranges. These port numbers are used to establish connections between the database server and the application processes or database utilities. These ports need to be opened through the firewall to make connections. The first available port in the list is used to make connection to the server. If no unused port is found, an error `NET_NO_UNUSED_PORT` is returned.

`VERSANT_SERVER_PORTS` is a comma separated list of port numbers or port ranges.

The following are the rules that apply for values that can be assigned to `VERSANT_SERVER_PORTS`:

- The value can contain digits between 0 to 9, comma(',') and hyphen('-')
- Any other character will be treated as an invalid value
- The hyphen has to be preceded and followed by a number, else the value will be treated as invalid
- If a range is specified, the start value has to be less than or equal to the end value

For E.g. 40010-4005 will be treated as an invalid value

- In any of the configuration files, the first space or tab is considered as the end of value specification. Thus, if the value contains space/tab then only the part prior to space/tab will be considered.

For E.g. : `VERSANT_SERVER_PORTS=40000,40005-40010,40026-40050,<tab>50000-50010`

In this case only "40000,40005-40010,40026-40050," will be considered, "50000-50010" will be ignored.

In case of invalid value for `VERSANT_SERVER_PORTS`, error `NET_EINVALIDPORTS` is thrown.

User may set the parameter `VERSANT_SERVER_PORTS` in any of the following places:

- In the user environment variable `VERSANT_SERVER_PORTS`.
- In a user configuration file set through `VERSANT_CFG`
- In the machine configuration file `-<windows directory>vrxxxyzz.ini`.  
Here `xxxyzz` indicates build version string.  
Example of valid file names are `vr701`, `vr070001`, `vr7001` etc.
- In the `VERSANT` system information file "`sysinfo`" in the `lib` directory.

The value for `VERSANT_SERVER_PORTS` will be searched in the following priority order:

1. Environment variables
2. User configuration file
3. Machine configuration file
4. System information file

If `VERSANT_SERVER_PORTS` is not defined, `VERSANT` will allow the operating system to assign any arbitrary port for communication with the application.

This configuration parameter should be used instead of `obe_port_begin` and `obe_port_end` parameters in the `profile.be` file of the database. The parameters `obe_port_begin` and `obe_port_end` are no longer supported.

**Example for setting `VERSANT_SERVER_PORTS`:**

**On running C Shell**

To set `VERSANT_SERVER_PORTS` in the `.login` file in the home directory:

```
setenv VERSANT_SERVER_PORTS 40000,40005-40010,40026-40050
```

### On running Bourne Shell

To set `VERSANT_SERVER_PORTS` in the `.profile` file in the home directory:

```
VERSANT_SERVER_PORTS = 40000,40005-40010,40026-40050
export VERSANT_SERVER_PORTS
```

## Windows Examples

### Personal computer example using normal parameters

Suppose you are running `WINDOWS` on a machine named `myMachine`, you are accessing another `WINDOWS` machine named `alpha` and a `UNIX` machine named `beta`, and the `osc-dbid` database system file is on `beta` in `/dbsys`.

To specify information about your machine, you must set:

```
VERSANT_ROOT          c:\versant
PATH                  c:\versant\7_0\bin;%PATH%;
LIBPATH               c:\versant\7_0\lib;%LIBPATH%
```

If you want to access software or databases on a remote machine, you must set:

```
VERSANT_USER          silas
```

If you want to create databases on your local machine, you must set:

```
VERSANT_DBID_NODE     beta
VERSANT_DBID           /dbsys
VERSANT_DB             c:\versant\mydb
```

After setting all of these variables, if you invoke `createdb` as:

```
createdb myDatabase
```

Then Versant will look at the following variables:

#### **PATH**

To find the `createdb` utility.

### **VERSANT\_DBID\_NODE**

To find the machine containing the `osc-dbid` file.

### **VERSANT\_DBID**

To find the directory containing the `osc-dbid` file.

### **VERSANT\_DB**

To find the database root directory.

After setting all of these variables, if you invoke `createdb` as:

```
createdb myDatabase@alpha
```

Then Versant will look at the following variable:

```
VERSANT_USER          To make a connection to alpha.
```

After making the connection, Versant will pass the `createdb` command to `alpha`. Machine `alpha` will then look to its own configuration parameters to decide where to find the `osc-dbid` file and where to place the new database.

### **Personal computer example using advanced parameters**

Suppose your `WINDOWS` machine is named `myMachine` and you are accessing another `WINDOWS` machine named `alpha` and a `UNIX` machine named `beta`, and the `osc-dbid` database system file is on `beta` in `/dbsys`.

If you have multiple releases of Versant on your local machine or databases created with multiple releases and want to assert the release number, you may optionally set:

```
VERSANT_REL            7.0.1
```

If you want to assert the location of the `osc-dbid` database system file to machines `alpha` and `beta`, you may optionally set:

```
VERSANT_DBID_NODE@alpha    beta
VERSANT_DBID@alpha         /dbsys
VERSANT_DBID_NODE@beta     beta
VERSANT_DBID@beta          /dbsys
```

If you want to assert the location of database root directories to machines `alpha` and `beta`, you may optionally set:

---

```
VERSANT_DB@alpha      d:\versant\db
VERSANT_DB@beta       /versant/db
```

If you have multiple releases of Versant on `alpha` and `beta` and databases created with multiple releases, to assert the release number and root directories, you may optionally set:

```
VERSANT_REL@alpha     7.0.1
VERSANT_ROOT@alpha    d:\versant
VERSANT_ROOT@beta     /usr/local/versant
VERSANT_REL@beta      6.0.5
```

After setting all of the above parameters, if you invoke `createdb` as:

```
createdb myDatabase@alpha
```

Then Versant will look at the following parameters:

## **VERSANT\_REL@alpha**

To assert the release number on `alpha`.

## **VERSANT\_ROOT@alpha**

To assert the software root directory on `alpha`.

## **VERSANT\_USER**

To find a login `username`.

## **VERSANT\_DBID\_NODE@alpha**

To assert the `osc-dbid` machine.

## **VERSANT\_DBID@alpha**

To assert the `osc-dbid` directory.

## **VERSANT\_DB@alpha**

To assert the database root directory.



---

This Chapter explains the “Error Codes” generated by Versant.

This Chapter covers the following in details:

- Error Handles, Numbers and Messages
- Virtual System Errors
- System Level Errors
- Versant Server Errors
- Versant Network Errors
- Versant Manager Errors
- Locking Errors
- Archive Errors
- Query Errors
- Version Errors
- Schema Errors
- Container Errors
- Event Notification Errors
- Virtual Attribute Errors

- System Utility Errors
- C++/Versant Errors
- GUI Tool Errors
- SQL & ST Errors
- Versant Statistics Errors



---

## ERROR HANDLES, NUMBERS AND MESSAGES

Categories of Errors      Series of Errors

Virtual System	0001-0699
System Level	0700-0999
VERSANT Server	1000-2999
VERSANT Network	3000-3999
VERSANT Manager	4000-4999
Locking	5000-5199
Archive	5200-5399
Query	5400-5599
Version	5600-5799
Schema	6000-6199
Container	6200-6399
Event Notification	6500-6540
Virtual Attribute	6600-6627
System Utility	7000-7999
	9000-9999
	11000-11999
	12300-12999
C++ Specific	8000-8999
	40000-40999
GUI Tool	10000-10999
SQL & ST	20000-20299
VERSANT Statistics	30000-34199

The format of the error code documentation is: **err number      handle      message**

### Incomplete Error Messages

If error numbers are displayed without an accompanying error message, then either your Versant installation is incomplete or needed environment variables have not been set correctly.

**For more information, please refer to the chapter "Installation" in the *Release Notes*.**

## VIRTUAL SYSTEM ERRORS, SERIES - 0001

0001	VSL_KEY_INVALID	Invalid Key with OS err #\$(oserr)
0002	VSL_BLKSZ_INVALID	Invalid block size
0003	VSL_BLKNO_INVALID	Invalid block number \$(block)
0004	VSL_SHM_ALLOC	Cannot allocate shared memory OS err #\$(oserr)

Versant cannot find enough shared memory for the server page cache.

If there are multiple databases running on this machine, shut one or more down (if possible), which will free the shared memory used by these databases. Otherwise you need to reconfigure your kernel so that it can allocate more shared memory resources on demand.

0005	VSL_SHM_ACCESS	Cannot access shared memory with OS err #\$(oserr)
0006	VSL_SHM_ELOCAT	Cannot locate shared memory with OS err #\$(oserr)

You can get this error while trying to connect to a database which is located on an NFS mounted directory visible to two different machines. If the database has been started by one machine and then you try to connect to it using another machine, you will get this error.

For example, suppose that a database named `db1` is visible on both machine `host1` and machine `host2`. Next, suppose that `db1` has been started by a connection from `host1`, which has created a server process, cleanup process, and a page cache in shared memory on `host1`. Now, if you try to connect to it from `host2`, you will get this error, because `host2` will try to start `db1` on `host2`, even though it is already running on `host1`.

If a database is in a directory visible to two machines, you can start it on either machine, as long as it has not already been started on the other machine. In the case of the above example where `db1` is already running on `host1` when you try to make a connection from `host2`, you could successfully make a connection if you used the syntax `db1@host1`, but your connection would fail if you used the syntax `db1` or `db1@host2`.

In general, you should always assign ownership of a database to a particular machine, typically configured as a server. Then, from client machines, always access the database with the syntax `database@server`. This will ensure that the database is started only on the server machine and that server processes and shared memory segments are present only on the server machine.

Ensuring that a database directory is visible only to the server machine will prevent this problem altogether, because it would make it necessary that all client machines access the database with `database@server syntax`.

0007	VSL_SHM_BUSY	Shared memory is in use with OS err #\$(oserr)
0008	VSL_SHM_ERM	Unable to remove shared memory with OS err #\$(oserr)
0009	VSL_CLK_ESTART	Timer already started
0010	VSL_CLK_NSTART	Timer not started
0011	VSL_CLK_ESYNC	Timer out of synchronization
0012	VSL_CLK_NBUF	No buffer specified
0013	VSL_CLK_ETIMDAT	Unable to obtain time of day
0014	VSL_PR_EPTBL	Process table invalid
0015	VSL_PR_NSLOT	Process table full

The process table mentioned here is a Versant internal structure. If you receive this message, you need to increase the transactions parameter in the `profile.be` file, stopping and restarting the database afterwards, until you no longer receive this message whilst running your application.

0016	VSL_PR_ECREAT	Cannot create process with OS err #\$(oserr)
0017	VSL_PR_NARGS	Too many arguments
0018	VSL_PR_EXEC	Process execution failed with OS err #\$(oserr)
0019	VSL_PR_NEXIST	Process #\$(pid) does not exist
0020	VSL_PR_EKILL	OS error #\$(oserr) in killing process #\$(pid)
0021	VSL_PR_EWAIT	OS error #\$(oserr) while waiting for process #\$(pid)
0022	VSL_PR_TBLBUSY	Process control block is in use
0023	VSL_SEM_EINIT	Latch not initialized
0024	VSL_SEM_EFREE	Latch already freed
0025	VSL_SEM_EGNCNT	Unable to get wait count
0026	VSL_SEM_ETIMEO	OS wait timeout err #\$(oserr) after waiting \$(timeout) seconds for latch hold by process #\$(pid)

0027	VSL_SEM_EWAIT	OS err #\$(oserr) with timeout set to \$(timeout) seconds while waiting for latch hold by process #\$(pid)
0028	VSL_SEM_ERETRY	Retry count #\$(count) exceeded while waiting for latch hold by process #\$(pid)
0029	VSL_SEM_BUSY	Latch is in use
0030	VSL_SEM_EPOST	OS err #\$(oserr) while posting latch
0031	VSL_SEM_ECREATE	OS err #\$(oserr) while creating latch

The transaction parameter in the database server profile file, `profile.be`, is set to a higher value than can be supported by your UNIX kernel. The solution is to increase the number of semaphores configured into your kernel.

For example, on Sun machines running SunOS, the kernel parameter of interest is `SEMMNS`, and the kernel configuration file should be in the file `/usr/kvm/sys/sun4c/conf`. See the documentation for your machine to determine how to increase the number of semaphores. Versant platform notes (which are part of the Release Notes) will contain default values which are appropriate for most purposes.

For example, the recommended kernel values for SunOS are,

options `SHMMNI=200`

options `SEMMNI=100`

options `SEMMNS=500`

The parameter `SEMMNI` indicates the maximum number of databases that can be active concurrently. The parameter `SEMMNS` specifies approximately the total number of database processes allowed for that system.

0032	VSL_SEM_ESETVAL	OS err #\$(oserr) while setting latch value
0033	VSL_SEM_EGETINFO	Error getting latch info
0034	VSL_SEM_ENTRIES	No more latch entries

More transactions are trying to access a database than the limit set by the transaction parameter for that database. The solution is to stop the database and increase the value of its transaction parameter in the server process profile file `profile.be`.

---

0035	VSL_SEM_UPGRADE	Attempt to upgrade latch
0036	VSL_SEM_NOLATCHES	No.of latches per execution unit exceeded
0037	VSL_SEM_DOWNGRADE	Attempt to downgrade latch failed
0041	VSL_VOL_ECLOSE	Unable to close volume with OS err #\$(oserr)
0042	VSL_VOL_EXISTS	Volume already exists OS err #\$(oserr)
0043	VSL_VOL_EREAD	Volume read OS error #\$(oserr)
0044	VSL_VOL_EWRITE	Volume write error OS err #\$(oserr)
0045	VSL_VOL_ESEEK	Seek OS error #\$(oserr) on volume
0046	VSL_VOL_EOPEN	Unable to open volume OS err #\$(oserr)
0047	VSL_VOL_ESTAT	OS err #\$(oserr) while trying to get volume status from dbpath '\$(file)'
0048	VSL_VOL_EDESTROY	Unable to destroy volume (\$(file)), OS error = #\$(oserr)
0049	VSL_VOL_INVALID	Invalid volume specified OS err #\$(oserr)
0050	VSL_EINPTR	Pointer is invalid
0051	VSL_EBUFSIZ	Buffer size is wrong
0052	VSL_EUSERNAM	Unable to get user name \$(string)

You are running NIS ("Yellow Pages") at your site, and the password files in /etc are different on the two machines. The solution is to run `ypmake` on the NIS server to propagate the /etc/passwd file to your client machine. On SGI machines that run NIS, your application has to be linked with the option `-lsun`.

0053	VSL_MEM_NOMEM	Out of process memory OS err #\$(oserr)
------	---------------	---

You need more swap space on your machine, at least as much swap space as the value you set for the parameter `swap_threshold` in your application process profile file. When you use Versant, the recommended swap space-to-RAM ratio is 3,1.

For more information refer to “swap\_threshold” on page 82, in "Chapter 3 - Database Profiles".

0054	VSL_SHM_EEXIST	Shared memory exists OS err #\$(oserr)
------	----------------	---

Versant needs to allocate shared memory for a database, but shared memory has already been allocated for that database or some earlier incarnation of that database. The solution is to stop the database with the `stopdb` utility, which will remove the shared memory segments.

For more information refer to “stopdb” on page 220, in "Chapter 4 - Database Utilities".

0055	VSL_PR_NPERM	No permission to kill process #\$(pid)
0061	VSL_FIXED_DEV_ZERO	Can't open "/dev/zero" with OS err err #\$(oserr)
0062	VSL_FIXED_MAP	Can't map memory with OS err #\$(oserr)
0063	VSL_FIXED_ADDRESS	Mapped at wrong address
0064	VSL_FIXED_UNMAP	Cannot unmap memory with OS err #\$(oserr)
0065	VSL_FIXED_CREATE	Cannot create segment with OS err #\$(oserr)
0066	VSL_FIXED_ATTACH	Cannot attach segment with OS err #\$(oserr)
0067	VSL_FIXED_DETACH	Cannot detach segment with OS err #\$(oserr)
0068	VSL_FIXED_REMOVE	Cannot remove segment with OS err #\$(oserr)
0071	VSL_EOSCPATH	Error while resolving configuration
0072	VSL_EDBPATH	No database path
0073	VSL_EDBIDPATH	No osc-dbid path
0074	VSL_EDBIDNODE	No osc-dbid node name
0075	VSL_ECONFIGFILE	Error in configuration file specified
0076	VSL_ELIBSYSINFO	Error in versant_root/lib/sysinfo
0077	VSL_EUNKNOWNKEYWORD	Unknown keyword
0078	VSL_ENOLOCALHOSTINFO	No Versant root found for local info

---

0079	VSL_ENOVERSANTROOT	No Versant root
0080	VSL_ESTRINGTOOLONG	Keyword or string too long
0081	VSL_EINVALIDVERS	Version number is invalid or non-numeric
0082	VSL_STAT_NBUF	No buffer specified
0083	VSL_STAT_EOSSTATS	Unable to obtain operating system statistics
0084	VSL_EAUTHPATH	No user authentication program path
0101	VSL_EXEC_ASYNC	Async EXEC of, %s failed, errno, %d
0102	VSL_EXEC_SYNC	Sync EXEC of, %s failed, errno, %d
0111	VSL_MSG_CANT_GET_QUEUE	Cannot get message queue
0112	VSL_MSG_CANT_SEND	Cannot send the message
0113	VSL_MSG_CANT_RECEIVE	Cannot receive the message
0114	VSL_MSG_CANT_CONTROL	Cannot manipulate the queue
0115	VSL_MSG_EXIST	Message queue exists
0121	VSL_MUT_EINIT	OS err #\$(oserr) while initializing a mutex
0122	VSL_MUT_ELOCK	OS err #\$(oserr) while locking a mutex
0123	VSL_MUT_EUNLOCK	OS err #\$(oserr) while unlocking a mutex
0124	VSL_MUT_EDESTROY	OS err #\$(oserr) while destroying a mutex
0125	VSL_TLS_ECREATE	Could not create tls key
0126	VSL_TLS_EGET	Could not get tls key
0127	VSL_TLS_ESET	Could not set tls key
0128	VSL_THR_ECREATE	Error in create new thread
0129	VSL_CV_EINIT	OS err #\$(oserr) while initializing a condition variable
0130	VSL_CV_ELOCK	OS err #\$(oserr) while locking a condition variable
0131	VSL_CV_EUNLOCK	OS err #\$(oserr) while unlocking a condition variable
0132	VSL_CV_EDESTROY	OS err #\$(oserr) while destroying a condition variable
0133	VSL_FILE_OPEN	OS err #\$(oserr) while opening file \$(file)

---

0134	VSL_FILE_WRITE	OS err #\$(oserr) while writing to file \$(file)
0135	VSL_FILE_MMAP	OS err #\$(oserr) while mapping a file to memory
0136	VSL_TLS_EDELETE	Could not delete tls key
0137	VSL_FILE_SEEK	OS err #\$(oserr) during a seek on file \$(file)
0138	VSL_FILE_READ	OS err #\$(oserr) while reading file \$(file)
0139	VSL_FILE_FSYNC	OS err #\$(oserr) while syncing file \$(file) to disk
0140	VSL_FILE_TRUNCATE	OS err #\$(oserr) while truncating file \$(file)
0141	VSL_FILE_EEXIST	File \$(file) does not exist
0142	VSL_FILE_EACCES	OS err #\$(oserr) while accessing the file \$(file)
0143	VSL_FILE_EXISTS	File \$(file) already exists OS err #\$(oserr)
0144	VSL_FILE_EOF	End of file detected while reading file
0145	VSL_FILE_UNABLE_TO_LOCK	Unable to lock file
0146	VSL_FILE_STAT	OS err #\$(oserr) during stat on file \$(file)
0147	VSL_THR_JOIN	OS err #\$(oserr) while waiting to join a thread
0148	VSL_TRACE_WRONG_VERSION	Incompatible trace file version. You may want to rename/remove the old trace file (typically .systrace file in database directory or entry against trace_file parameter in profile.be).
0149	VSL_OSERR	OS function '\$(osfunc)' failed with err '\$(oserr)'
0150	VSL_DL_OPENFAILED	Cannot open Library for loading
0151	VSL_DL_CLOSEFAILED	Cannot close Library for loading
0152	VSL_DL_INVALIDHANDLE	Invalid Handle
0153	VSL_DL_INVALIDSYM	Invalid Symbol
0154	VSL_PASSWD_EXCEEDSSIZE	Password size exceeds 256



---

0155	VSL_CANNOT_GET_PASSWD	Cannot get password from terminal with OS err #\$(oserr)
0156	VSL_RAWDEV_DISK_DESC_TYPE	Device description error

You may want to rename/remove the old trace file (typically `.systrace` file in database directory or entry against `trace_file` parameter in `profile.be`).

0157	VSL_DIR_DELETE_FAILED	OS err #\$(oserr) while deleting directory
0158	VSL_DIR_NOT_A_DIRECTORY	The specified path is not a directory
0159	VSL_DIR_NOACCESS	Database directory not found or no access permission

## SYSTEM LEVEL ERRORS, SERIES - 0700

0700	SL_EXPR_UNSUPPORTED_UNARY_OP	This unary operator is not supported
0701	SL_EXPR_UNSUPPORTED_BINARY_OP	This binary operator is not supported
0702	SL_EXPR_UNSUPPORTED_COND_OP	This conditional operator is not supported
0710	SL_EXPR_UNSUPPORTED_CONST_TYPE	This constant type is not supported in expressions
0711	SL_EXPR_UNSUPPORTED_ATTR_TYPE	This attribute type is not supported in expressions
0715	SL_EXPR_BAD_OPERAND_TYPE	Data type of operand is not compatible with the operator
0730	SL_EXPR_WOULD_OVERFLOW	The evaluation of the expression would cause a numeric overflow
0731	SL_EXPR_WOULD_UNDERFLOW	The evaluation of the expression would cause a numeric underflow
0732	SL_EXPR_DIVIDE_BY_ZERO	The evaluation of the expression would cause a divide by zero
0792	SL_BAD_SDA_BASE_ENV	Could not parse VERSANT_SDA_BASE environment variable
0800	SL_DATABUF_BAD_START_SIZE	Invalid start size
0801	SL_DATABUF_NOMEM	Memory not available
0802	SL_DATABUF_NOT_INITIALIZED	Data buffer is not initialized
0803	SL_DATABUF_BAD_DATA_LENGTH	Invalid data length
0804	SL_DATABUF_MAX_SIZE_REACHED	Data buffer has already reached its maximum size
0805	SL_DATABUF_NULL_DATA_POINTER	Null data pointer
0806	SL_DATABUF_FIXED_SIZE_MISMATCH	Data did not fit in fixed size data buffer
0900	SL_MEM_INVALID_SIZE	Invalid size specified
0902	SL_MEM_NOELEMENT	Out of elements
0903	SL_MEM_NOMEM	Out of heap memory
0911	SL_BAD_LIST	Bad list
0912	SL_BAD_ITEM	Bad list element
0913	SL_ITEM_NOT_FOUND	Item not found in the list
0914	SL_BAD_VSTR	Bad vstr (variable length string)
0915	SL_BAD_BUFFER	Bad buffer

---

0916	SL_BUFFER_TOOSMALL	Buffer is too small
0931	SL_MSG_WOULDBLOCK	Message would have blocked
0932	SL_MSG_ALONE	No frontend shared memory
0941	SL_PATH_NOHOME	No home environment
0942	SL_DBNAME_TOOLONG	Database name too long
0943	SL_USERNAME_TOOLONG	User name too long
0944	SL_NODENAME_TOOLONG	Node name too long
0950	SL_CANT_CREATE_LOCAL	Cannot create heap in local memory
0951	SL_CANT_CREATE_SHARED	Cannot create heap in shared memory
0952	SL_CANT_EXTEND_LOCAL	Cannot extend heap in local memory

You are running out of application memory, which can occur if there are too many processes running on your machine or if your application is manipulating too many objects in a transaction. First try reducing the number of processes. One thing you can do is stop any databases that are running but not being used.

On UNIX, you can check for unused databases with the following:

```
ps aux | grep cleanbe
```

You can reduce the memory requirements of your application by committing more often and/or using pin regions to hold only necessary objects in memory.

0953	SL_CANT_EXTEND_SHARED	Cannot extend heap in shared memory
0954	SL_HEAP_MAXIMUM_SIZE	Heap hit specified maximum size
0955	SL_CANT_ATTACH_CONTIGUOUS	Cannot attach shared memory contiguously
0957	SL_CANT_ATTACH_RIGHT	Cannot attach shared memory at the right address
0958	SL_SEGMENT_TABLE_FULL	Shared memory segment table is full
0960	SL_BAD_SHARED_MAGIC	Bad shared memory format
0961	SL_WRONG_SDA_VERSION	Incompatible version of SDA
0962	SL_WRONG_ODA_VERSION	Incompatible version of ODA
0963	SL_CANT_CREATE_KEY	Cannot create session key file
0964	SL_ATTACH_EXTEND_SEG	Cannot attach extension segments

The machine containing a database has run out of shared memory segments. The solution is to reconfigure the operating system on the machine containing the database so that it can support a larger number of shared memory segments.

0965	SL_CANT_RELEASE _LATCH	Cannot release latch
0966	SL_CANT_GET_LATCH	Cannot get latch
0967	SL_CANT_ALLOC_PAGE	Cannot alloc page space
0968	SL_INVALID_HEAP_ADDRESS	Invalid pointer in heap function
0969	SL_AREA_ALREADY_FREE	Area is marked as already free
0970	SL_BAD_PAGE_STATUS	Page status is bad
0971	SL_PANIC_MESSAGE	[E%d, %s] Versant process terminated in file %s at line %d
0972	SL_DB_PANIC_MESSAGE	[E%d, %s] Versant panicking; SDA has crashed; database shutting down.
0973	SL_CANT_GET_LATCH_TWICE	Cannot get the same latch twice
0974	SL_BUFFER_READ	Buffer Read
0975	SL_VERSION_MISMATCH	Database is \$(db_version). Versant is \$(versant_version). Use different version of versant or run convertldb
0976	SL_CRASH_NOTALLOWED	Backend crash test not allowed
0977	SL_MESSAGE_TEST	Test message functionality \$(loid) \$(filename) \$(4byte) \$(u4byte) \$(oserr)
0978	SL_MESSAGE_TRACE	Trace along error path
0979	SL_BAD_INPUT	Input parameter is bad
0980	SL_DIV_ZERO	Trying to divide by zero
0981	SL_CPI_KEYNOTFOUND	Can't find an entry with this plugin-typename
0982	SL_CPI_CANNOTLOAD	Can't load the plugin
0983	SL_CPI_INVALIDID	This plugin not supported
0984	SL_VCP_ERROR	Error in the Plugin Module
0985	SL_CPI_PATHEXCEEDSSIZE	Plugin path exceeds max-size
0986	SL_CPI_INVALIDDLLNAME	Not a valid plugin name
0987	SL_CPI_NAMEEXCEEDSSIZE	Plugin_typename exceeds limit. ( max, 32 chars)
0988	SL_CPI_NULLPTR	NULL pointer

---

0989	SL_CPI_PROFILEEXCEEDSLIMIT	Only 8 Entries allowed
0990	SL_ER_DOUBLEBUFFER_INVALIDSIZE	Invalid Buffer Size
0991	SL_ER_DOUBLEBUFFER_EMPTY	Buffer Empty
0992	SL_CHARACTER_SET_UNSUPPORTED	Specified character set not supported

## VERSANT SERVER ERRORS, SERIES – 1000, 2000

1000	SM_E_UNKNOWN	Unknown error
1001	SM_E_INTERNAL	Internal SM error
1002	SM_E_BAD_NODE_OPER	Invalid evaluation operator
1003	SM_E_BAD_PARAM	Invalid internal function parameter
1004	SM_E_TOO_MANY_BOOLEANS	Too many boolean terms
1006	SM_E_BAD_COMPARATOR	Invalid comparison operator
1007	SM_E_BAD_USER_PARAM	Invalid user input parameter
1008	SM_E_BAD_ACCESS_MODE	Invalid access mode
1009	SM_E_KEY_NOT_FOUND	Key \$(key) cannot be found
1010	SM_E_BAD_CSR_POSITION	Invalid cursor position
1011	SM_E_BAD_DOMAIN_TYPE	Invalid domain specified for attribute
1013	SM_E_TOO_MANY_INDEXES	Too many indexes
1014	SM_E_INDEX_NOT_FOUND	Index cannot be found
1015	SM_E_BAD_CURSOR_TYPE	Invalid cursor type
1017	SM_E_BAD_ATT_TYPE	Invalid attribute type
1018	SM_E_BAD_NAME_LENGTH	Invalid attribute type
1019	SM_E_VOL_NOT_FOUND	Volume not found
1020	SM_E_NDX_NOT_FOUND	Index not found
1021	SM_E_CLS_NOT_FOUND	Class not found
1027	SM_E_CSR_NOT_OPENED	Cursor is not open
1028	SM_E_ID_NOT_FOUND	Id not found in the catalog
1030	SM_E_NO_SUCH_OBJECT	No such object, loid = \$(loid)
1031	SM_E_BAD_NDX_ID	Bad index id
1032	SM_E_BAD_CSR_ID	Invalid cursor id
1033	SM_E_OUT_OF_CSR_SPACE	Out of cursor space
1034	SM_E_BAD_OPER	Invalid evaluation operator
1035	SM_E_BAD_READ_OFFSET	Invalid read offset
1036	SM_E_BAD_READ_LEN	Invalid read length
1037	SM_E_NO_SPACE	No space
1038	SM_E_NO_SYS_SEGMENTS	System segments not found
1039	SM_E_REMOVE_AT_SEG	Error removing AT segment
1040	SM_E_ID_EXISTS	Id already exists

---

1041	SM_E_OBJID_NOT_FOUND	Object id not found
1042	SM_E_SEG_EXISTS	Segment already exists
1043	SM_E_SEG_NOT_FOUND	Segment not found
1044	SM_E_SEG_NOT_EMPTY	Segment not empty
1045	SM_E_CLS_EXISTS	Class already exists
1046	SM_E_CLS_NOT_EMPTY	Class not empty
1047	SM_E_BAD_BUFFER_SIZE	Buffer size too small
1048	SM_E_BAD_SYS_SEG_ID	Invalid system segment id
1049	SM_E_NAME_EXISTS	Name already exists
1050	SM_E_BAD_SEG_ID	Invalid segment id
1051	SM_E_BAD_CLS_ID	Invalid class id
1053	SM_E_BAD_CLS_NAME	Invalid class name
1054	SM_E_REMOVE_SYS_CLS	Error removing system class
1055	SM_E_REMOVE_SYS_SEG	Error removing system segment
1056	SM_E_BAD_SEG_NAME	Invalid segment name
1057	SM_E_BAD_SYS_CLS_ID	Invalid sys class id
1064	SM_E_BAD_CLS_TYPE	Invalid class type
1068	SM_E_ATT_NOT_FOUND	No entry in attribute-segment
1071	SM_E_IDS_NOT_FOUND	No entry in id segment
1072	SM_E_SEGMENT_REFCNT	Invalidate segment reference count (0)
1073	SM_E_SEGMENT_NOT_FOUND	No entry in segment-segment
1074	SM_E_SEGMENT_OWNERCNT	Segment owner count error
1075	SM_E_SEGMENT_OPEN_ERROR	Segment open error
1076	SM_E_NDXSEG_NOT_FOUND	Index segment not found
1077	SM_E_MAX_OPENFILES	Number of open files already maximum
1078	SM_E_CLS_SEG_NOT_FOUND	Class segment not found
1079	SM_E_NDX_SEG_NOT_FOUND	Index segment not found
1080	SM_E_BAD_NDX_TYPE	Wrong index type
1081	SM_E_BAD_MO_TAILINFO	Inconsistent medium object tail information
1083	SM_E_OUT_OF_VOL_SPACE	All volumes exhausted

The database configuration parameter `sysvol` in the server process profile file determines the initial size of the database when it is created, and changing it after a database has been created will have no effect

**For more information refer to “Database Creation Basics” on page 54, in "Chapter 2 - Database Creation".**

1099	SM_E_SDA_CRASHED	SDA has crashed
1100	SM_E_NOMEM	Out of backend heap memory
1102	SM_E_FATAL	Storage manager encountered fatal error %d at file %s: line %d
1103	SM_E_SDA_INIT	Error in Storage manage shared memory initialization
1104	SM_E_CNVN_UNDERFLOW	Key conversion underflow
1105	SM_E_CNVN_OVERFLOW	Key conversion overflow
1106	SM_E_CNVN_WRONG _TYPE	Invalid type for key conversion
1107	SM_E_INCOMPAT_KEY_TYPE	Incompatible key type
1108	SM_E_MULTIVALUED_PATH	Multi-valued path attrs are not allowed here
1109	SM_E_INCOMPLETE_PATH	Multi-valued path attrs are not allowed here
2500	SM_LOG_INVALID_PARM	Invalid parameter to logging interface
2501	SM_LOG_NO_ACTIVE_TR	No active transaction
2502	SM_LOG_FATAL	Fatal unrecoverable error
2503	SM_LOG_CANNOT_EXTEND	Raw device, can not extend log file
2504	SM_LOG_NO_LOG_ADR	Invalid log record address
2505	SM_LOG_NO_PLOG_SPACE	Out of physical log file space
2506	SM_LOG_NO_LLOG_SPACE	Out of logical log file space
2507	SM_LOG_NO_PLOG_FILE	No/invalid physical log volume
2508	SM_LOG_NO_LLOG_FILE	No/invalid logical log volume
2509	SM_LOG_NOPACKSPACE	Unable to re-organize/pack file
2510	SM_LOG_INVALID_PLOG	Invalid physical log version, physical recovery disabled
2511	SM_LOG_INVALID_LLOG	Invalid logical log version, logical recovery disabled
2512	SM_LOG_INCOMPAT_VERSION	Log file and utility versions are incompatible
2513	SM_LOG_BAD_BGN_REC	Deferred begin record is wrong
2514	SM_LOG_BAD_LOGPTR	Log Pointer corrupted
2515	SM_LOG_NO_PARENT_TR	No parent log pointer



---

2516	SM_LOG_OVERFLOW	Number overflow
2517	SM_LOG_MEM_CORRUPTED	Log memory corrupted
2518	SM_LOG_FILE_CORRUPTED	Log file corrupted
2519	SM_LOG_UIDTOOSMALL	Invalid uid
2520	SM_LOG_DISABLE_CP	Error in disable log CP
2521	SM_LOG_PLOG_NOT_EMPTY	Plog file not empty
2522	SM_LOG_LLOG_NOT_EMPTY	Llog file not empty
2523	SM_LOG_CANNOT_EXTEND_PLOG	Plog cannot be extended. Either there is insufficient space in the filesystem/device, or, PLOG has already reached its maximum size
2524	SM_LOG_CANNOT_EXTEND_LLOG	Llog cannot be extended. Either there is insufficient space in the filesystem /device, or LLOG has already reached its maximum size.
2525	SM_LOG_WRONG_DB_VERSION	DB version not right, need to run convertdb
2526	SM_LOG_NOT_EMPTY	Log files have to emptied by stopdb and startdb in order to run convertdb
2527	SM_LOG_END_OF_UNIT	End of log unit reached
2701	SM_E_BADCURSOR	Bad scan cursor
2702	SM_E_BADSCANTYPE	Scan type incompatible with operation
2703	SM_E_BADRELOCATION	Incorrect relocation
2704	SM_E_NOFILENO	File number not found
2705	SM_E_BADFILENO	Invalid file number
2706	SM_E_BADSCANID	Invalid scan id
2707	SM_E_SCANFILENOTMATCH	Scan id does not match file number
2708	SM_E_NOMOREMEMORY3	No more memory space for scan table
2709	SM_E_ACCESSVIOLATION	Attempt to update a read file
2710	SM_E_NONEXTOID3	No next object id
2711	SM_E_NOPREVOID3	No previous object id
2712	SM_E_PAGENOTFOUND	Page not found in buffer pool
2713	SM_E_NOFREEBUFFERS	No more free buffers
2714	SM_E_WRONGBUFFER	Accessing the wrong buffer

---

2715	SM_E_NULLFIDPARM	Null file id parameter
2716	SM_E_NULLPIDPARM	Null page id parameter
2717	SM_E_BADMODEPARM	Bad file mode parameter
2718	SM_E_INVTIMESTAMP	Timestamp is invalid
2719	SM_E_NONEXTKEY	No next key in index
2720	SM_E_NOPREVKEY	No previous key in index
2721	SM_E_NOFIRSTINDEX	No first key in index
2722	SM_E_NOLASTINDEX	No last key in index
2723	SM_E_NONEXTOID2	Current object id is the last for this key
2724	SM_E_NOPREVOID2	Current object id is the first for this key
2725	SM_E_DUPLICATEKEY	Duplicate key \$(key) found
2726	SM_E_DUPLICATEKEYPTR	Duplicate key found in B-tree nodes
2727	SM_E_KEYNOTFOUND	Cannot find the key
2728	SM_E_INDEXNUMTOOLARGE	Index number too large (create index)
2729	SM_E_KEYLENGHTOOLONG	Index Key length is too long
2730	SM_E_BADSLOTCOUNT	Negative slot count
2731	SM_E_KEYALREADYEXISTS	Key \$(key) already exists
2732	SM_E_FILENAMEETOOLONG	File name too long
2733	SM_E_NOOIDMATCH	The index to be deleted is not found
2734	SM_E_ILLEGALCURSOR	Bad cursor for index scan
2735	SM_E_VOLNOTOPEN	Volume not open
2736	SM_E_NOSUCHFILE	No file by that name
2737	SM_E_VOLALREADYOPEN	Volume already open
2738	SM_E_FILEALREADYEXISTS	File already exists
2739	SM_E_NOMOREMEMORY2	Failed to allocate memory
2740	SM_E_TOOMANYVOLUMES	Too many volumes currently in use
2741	SM_E_FILESTILLOPEN	Attempt to remove a currently opened file
2742	SM_E_BADOPENFILENUM	Open file id is out of range
2743	SM_E_WRONGUSER	This open file belongs to another user
2744	SM_E_TOOMANYOPENFILES	Too many open files
2745	SM_E_UNKNOWNMODE	Unknown access mode

---

2746	SM_E_MODECONFLICT	Access modes conflict
2747	SM_E_UNKNOWNPROT	Unknown protection mode
2748	SM_E_PERMISSIONDENIED	File permission denied
2749	SM_E_NOPERMISSION	No file permission
2750	SM_E_DEVSEEKERROR	Device seek error
2751	SM_E_DEVREADERROR	Device read error
2752	SM_E_DEVWRITEERROR	Device write error

This means that a disk write has failed, probably because the disk is full.

2753	SM_E_MOUNTFAILED	Device mount failed
2754	SM_E_DISMOUNTFAILED	Device dismount failed
2755	SM_E_VOLNOTMOUNTED	Referenced volume not mounted
2756	SM_E_TOOMANYVOLS	Too many volumes mounted
2757	SM_E_TOOMANYFILES	Too many files created
2758	SM_E_NOSPACEONDISK	No space on system disk
2759	SM_E_NOMOREMEMORY0	Insufficient main memory
2760	SM_E_FIDPIDNOTMATCH	Volume IDs in the file id and page id are inconsistent
2761	SM_E_FILENOTINUSE	Reference a non-existent file
2762	SM_E_INVALIDPID	Invalid page number, pid = \$(pid)
2763	SM_E_INVALIDFID	Invalid file number
2764	SM_E_BADHEADER	Bad volume header
2765	SM_E_NULLPIDPTR0	Null page id pointer
2766	SM_E_NULLFIDPTR0	Null file id pointer
2767	SM_E_NULLBUFPTR	Null memory buffer pointer
2768	SM_E_NAMEINUSE	File name already in use
2769	SM_E_BADSLOTNUMBER	Invalid slot number
2770	SM_E_LISTOFEMPTYSLOTS	Empty slots at end of page
2771	SM_E_NOROOMONPAGE	No space for expansion on page
2772	SM_E_NULLOIDPTR	Null object id pointer
2773	SM_E_NULLPIDPTR2	Null page id pointer
2774	SM_E_NULLFIDPTR2	Null file id pointer
2775	SM_E_NULLPAGEPTR	Null page buffer pointer
2776	SM_E_NULLOBJADDR	Null object buffer address
2777	SM_E_DELNOEXTOBJ	Deleting a non-existing object
2778	SM_E_PAGENOTINFILE	Referenced page not in file
2779	SM_E_ENDOFFILE	End of file reached

---

2780	SM_E_OBJWONTFIT	No space for object on this page
2781	SM_E_CANTINSERTOBJ	Cannot insert object for unknown reason
2782	SM_E_VOLUMESTILLACTIVE	Attempt to dismount a volume with open files
2783	SM_E_NOMORESICES	Attempt to grow a long data item beyond limit
2784	SM_E_BADDATATYPE	Bad data type detected
2785	SM_E_ILLEGALOP	Illegal operator found
2786	SM_E_VOLMOUNTED	Volume already mounted
2787	SM_E_VOLEXISTS	Volume already exists
2788	SM_E_CREATFAILED	Volume create failed
2790	SM_E_VAERROR	VA internal error
2800	SM_E_INVALID_PASSWORD	Invalid Password specified
2801	SM_E_CANNOT_CHANGE_PASSWORD	Cannot Change Password
2802	SM_E_NULL_PASSWORD	No Password
2803	SM_E_INVALID_USER	The user has no password and is not logged into the OS as that user
2901	SM_LOCK_WOULDBLOCK	Lock on \$(loid) would have blocked process
2902	SM_LOCK_DEADLOCK	Deadlock detected on attempt to lock \$(loid)
2903	SM_LOCK_TIMEDOUT	Lock wait Timed out on \$(loid)
2904	SM_LOCK_DELETED	Object deleted
2905	SM_LOCK_NOUPGRADE	Lock upgrade not allowed
2906	SM_LOCK_INVALID	Invalid lock mode
2907	SM_LOCK_MODLM	Lock model modified
2908	SM_LOCK_LMNOTCHANGED	Cannot change lock model
2909	SM_LOCK_NOTSUPER	Not a superset of Versant Standard Lock Model
2910	SM_LOCK_NOLKONOBJ	Found no lock on obj
2911	SM_LOCK_NO_EXCACC	Not holding exclusive access
2912	SM_LOCK_OWEN_EXCACC	Own exclusive access
2913	SM_LOCK_TS_CHANGED	Object may have changed
2921	SM_TR_EXISTS	Transaction already exists
2922	SM_TR_FAIL	Transaction failed
2923	SM_TR_NOTFOUND	Transaction not found

---

2924	SM_TR_NO_LOG	No log => no save point
2925	SM_TR_OBJ_NOTFOUND	Object isn't in its hash chain
2926	SM_TR_MULTIPLE_LOCKS	Multiple locks on the same transaction and object
2927	SM_TR_COMPATIBILITY	Unknown relation between transactions
2928	SM_TR_LOG_RECORD_TYPE	Unknown log record type
2929	SM_TR_CANT_LOCK_OBJECT	Cannot lock object
2930	SM_TR_WRONG_COUNT	Number of to be recovered transaction changed
2931	SM_TR_RECOVERING	Recovery of distributed transactions in progress
2932	SM_TR_ALREADY_ATTACHED	XA transaction already active in backend
2933	SM_TR_NEW_XACTS_BLOCKED	Database being stopped; new transactions are blocked
2934	SM_TR_UPDATES_BLOCKED	Database being stopped; new updates are blocked
2935	SM_TR_ADDVOL_BLOCKED	Another addvol is active; addvols is blocked
2936	SM_TR_NOT_ZOMBIE	Transaction is not a dead/zombie transaction
2937	SM_TR_INVALID_STATE	Transaction is in the wrong state
2961	SM_SHM_NOT_READY	SDA not ready
2962	SM_SHM_SHUTTING_DOWN	SDA is shutting down
2971	SM_DB_IS_READONLY	Database is read-only mode
2972	SM_DB_IS_ACTIVE	Active transaction in database
2973	SM_DB_IS_REPLICATED	Replicated database
2981	SM_PS_NOT_FOUND	Process not found
2982	SM_CANT_FLUSH_BUFFER	Page buffer not flushable
2983	SM_CANT_GET_CONFIG_INFO	Unable to obtain configuration info
2984	SM_KILL_FE_CLEANUP	Killing the front end cleanup process[%d]
2985	SM_KILL_BE_CLEANUP	Killing the back end cleanup process[%d]
2986	SM_KILL_PROCESS	Killing process[%d]
2987	SM_DBA_ONLY	Only the DBA can access the database currently

2988	SM_DBA_NOT_ALONE	Only one process can access the database
2989	SM_DBA_MODE_SET	DBA-only mode is already set
2990	SM_DB_IS_UNSTARTABLE	Database is unstartable. Use dbinfo to change mode.
2991	SM_UNSTARTABLE_MODE_SET	Unstartable mode is already set
2994	SM_NOT_IN_USER_LIST	Not in userlist. Use DBUSER to add.
2995	SM_E_SAMEKEYFOUND	Identical key not allowed on unique index, key = \${key}
2996	SM_E_OUTOF_CACHE_MEMORY	No BufferPool Mem Available
2997	SM_E_INVALIDBUF	Buffer not useful for reading
2998	SM_E_BUF_WRONGSIZE	Buffer with wrong size

## VERSANT NETWORK ERRORS, SERIES - 3000

3000	NET_EINPTR	Invalid pointer (condef)
3001	NET_ETIMEO	Network layer timeout
3002	NET_ENOMEM	Unable to allocate memory
3003	NET_ENOPN	Connection is not opened
3004	NET_EREAD	Network layer read error

There has been a network error.

This error can occur if the remote `inetd` daemon is not able to start the `ss.d` process. In this case, the solution is to update the `/etc/inetd.conf` file with the correct entry and then restart the `inetd` daemon. For further information, see the installation instructions in the Release Notes.

This error can also occur if the `ss.d` process is not able to start the server process for the database. This can occur in the following situations,

- if the Versant machine configuration file `/etc/.oscxxyynn` (where `xxyynn` corresponds to your release number) does not have the right setting for `VERSANT_ROOT` (which specifies the software root directory),
- if different releases are running on the client and server and the configuration parameter `VERSANT_REL@node` has not been properly set on the client machine, or
- if the number of the Versant release running on the server is not numerically equal to or greater than the client release number.

This error can also occur if the database server process crashes while the application process was communicating with it, in which case there will be a core dump. This could happen due to resource limitations such as lack of shared memory or semaphores. In this case, the cause is normally written to a file called `LOGFILE` under the Versant database directory. If `LOGFILE` does not contain the information you need to solve the problem, please call Versant Technical Support.

**For more information, on the `.oscxxyynn` machine configuration file, the `VERSANT_ROOT`, `VERSANT_REL` configuration parameters, and the `LOGFILE` error file, refer to Chapter 5 “Configuration Parameters” on page 315.**

3005	NET_EWRITE	Network layer write error
------	------------	---------------------------

The database server process has crashed while the application process was communicating with it, in which case there will be a core dump. This could happen due to resource limitations such as lack of shared memory or semaphores. The cause is normally written to the file `LOGFILE` under the

Versant database directory. If `LOGFILE` does not contain the information you need to solve the problem, please call Versant Technical Support.

3006	<code>NET_ENODNAM</code>	Unable to obtain node name
3007	<code>NET_INVREQ</code>	Invalid RPC function
3008	<code>NET_INVHOST</code>	Invalid host name <code>\$(string)</code> , OS error = <code>\$(oserr)</code>
3009	<code>NET_ECONNECT</code>	Low level connect error

A network error has occurred.

This error can occur if the network `inetd` process does not know how to start the `oscssd` (`ss.d`) server process for a database. In this case, the solution is usually to correct the entry for the `oscssd` process in the `/etc/inetd.conf` configuration file and then killing the `inetd` process with `kill -1`. For further information, see the installation notes in the Release Notes.

This error can also occur if the database server process crashes while the client application process was communicating with it, in which case there will be a core dump. In this case, examine the `LOGFILE` error file in the database directory to see if you can find the cause of the problem. If you cannot find the problem in `LOGFILE`, please contact Versant Technical Support.

**For more information refer to “LOGFILE” on page 297, in “Chapter 5 - Directories and Files”.**

Another cause for this error is that your system has run out of socket descriptors. Socket descriptors and file descriptors are allocated from the same pool. In this case, you need to increase the number of file descriptors configured into your kernel. Since, the number of file descriptors is usually dependent on the number of concurrent users supported by your kernel, it should be sufficient to increase the parameter that sets the maximum number of users. This parameter is usually called `MAXUSER`. For information on setting the number of users, see your operating system manuals.

3010	<code>NET_TCP_ESOCK</code>	Error obtaining socket
3011	<code>NET_INVUSRNM</code>	Invalid user name/password string <code>\$(string)</code>

A login error has occurred.

If this error occurred on a personal computer, it may mean that the configuration parameter `VERSANT_USER` has not been set to a valid user of the database involved. In this case, set



VERSANT\_USER to a valid user name or have the DBA for the database add the current user name to the list of authorized user.

This error may also mean that your user name is not a valid login name on the remote machine. In this case, have the system administrator for the remote machine to add your user name to the list of valid login names on the machine where the database is located.

3012	NET_INVCONSTR	Invalid connection string
3013	NET_ENOSTREAM	Expected stream to follow
3014	NET_INVREQ2	Invalid request (lower level)
3015	NET_EPCREAT	Unable to start local OBE
3016	NET_ECONLOST	Network connection lost
3017	NET_E2MANYOBJ	Read/write object count exceeded
3018	NET_ENOOSGRP	OSC_GROUP not on host
3019	NET_NULLDRV	Null driver called
3020	NET_INVSEVR	Invalid service (\$(string)), OS err #\$(oserr)

There is a Yellow Pages configuration error. This error happens if your site is running NIS "Yellow Pages" and the Versant oscssd 5019/tcp entry in your /etc/services file does not also exist on the NIS server machine. This solution is to update the /etc/services file on the NIS server with the entry oscssd 5019/tcp. For information, see the installation chapter in the Release Notes.

If you are not able to get access to the NIS server, you can alternately disable the NIS client software on your machine. One way to disable NIS is by doing the following,

1. Move the ypbind file, :- mv /usr/etc/ypbind /usr/etc/ypbind.orig.
2. Reboot.

Once you disable the NIS client software on your machine, you should be able to run Versant successfully.

3021	NET_PURGE_STREAM	Unable to purge stream
3022	NET_INVSEVR_INFO	Invalid server process info
3023	NET_NETWORK_TYPE	Invalid network type for server
3024	NET_NETINIT	Unable to init server network
3025	NET_INTERNAL	Internal error
3026	NET_INITMSG	Unable to get initial message
3027	NET_CLIENT_NAME	Unable to get peer name of client
3029	NET_ROOTPATH	Unable to obtain root path
3030	NET_EXECSERV	Unable to exec backend server

3031	NET_NOT_IN_USERLIST	Not in userlist. Use DBUSER to add.
3032	NET_NO_POLLING_YET	Polling has not come in yet
3033	NET_OBSOLETE	RPC is no longer supported.
3034	NET_UNIMPLEMENTED	RPC is not yet implemented
3035	NET_EACCESS	User authentication failed
3036	NET_RW_ACCESS_REQUIRED	Read/Write access is required
3037	NET_THREADS_REQUIRED	You must link with the threads library to use lp connections
3038	NET_NO_UNUSED_PORT	Cannot find an unused port, all ports specified in VERSANT_SERVER_PORTS are in use.
3039	NET_XID_CONVERSION_ERR	Convert xid from 32/64 bit to 64/32 bit failed
3040	NET_RETRY	Local server did not receive a response from an authoritative server. Try again later, OS err #\$(oserr)
3041	NET_NO_RECOVERY	An unrecoverable error occurred, OS err #\$(oserr)
3042	NET_NO_ADDRESS	The hostname (\$(string)) is valid but does not have an Internet address at the name server, OS err #\$(oserr)
3043	NET_SERVICE_UNAVAILABLE	None of the name services configured are running or available, OS err #\$(oserr)
3044	NET_NO_NAME	The requested address parameter (\$(string)) is valid but does not have a name at the name server, OS err #\$(oserr)
3045	NET_EINVALIDPORTS	VERSANT_SERVER_PORTS contains invalid value.

## VERSANT MANAGER ERRORS, SERIES - 4000

4000	OM_SE_NOT_FOUND	Not in the session
4001	OM_SE_ALREADY_IN	Already in a session
4002	OM_SE_NAME_EXISTS	Session name exists
4003	OM_SE_INVAL	Invalid argument
4004	OM_SE_NOT_PARENT	Not a parent process
4005	OM_SE_PARENT	Parent process
4006	OM_SE_NO_SESSION	No session to join
4007	OM_SE_NO_JOIN	No dynamic join in
4008	OM_SE_INVAL_USERNAME	Invalid username
4009	OM_SE_INVAL_SESSNAME	Invalid session name

You are using an invalid session name in your begin session routine. Session names cannot exceed 31 characters.

**For more information, on other rules related to session names, please refer to the chapter “Sessions” in the *Versant Database Fundamentals Manual*.**

4010	OM_SE_NOT_IN_THREAD	o_bginthread must be called
4011	OM_SE_IN_THREAD	o_bginthread already called
4012	OM_SE_NOT_IN_SESSION	o_setsession must be called
4013	OM_SE_NOT_IN_TABLE	Invalid session specified
4014	OM_SE_INVALID_PID	Invalid process id
4015	OM_SE_NOT_THREAD_SAFE	O_THREAD option required
4017	OM_SE_THREADS_REQUIRED	You must link with the threads library to use threaded sessions
4018	OM_BAD_OPTIONS	Invalid options specified by user
4019	OM_SE_SESSION_IN_USE	Session is in use by other threads
4020	OM_SE_FE_LOCKFILE_ERR	Error opening the FE lock file
4021	OM_TR_NOT_EXISTS	No such transaction
4024	OM_TR_EXISTS	Transaction already exists
4025	OM_TR_INVAL_ACTION	Invalid action
4026	OM_TR_NO_LOGGING	Logging not turned on
4027	OM_TR_IS_ABORTABLE	Current transaction is abortable
4028	OM_TR_JOIN_IMPLICIT	Cannot end transaction implicitly started by o_joinsession.

---

4029	OM_TR_BEGIN_IMPLICIT	Cannot end transaction implicitly started by o_beginsession.
4030	OM_TR_IMPLICIT	Cannot end implicitly started transaction.
4031	OM_TR_PARENT_PROC	Cannot end a transaction in the parent process.
4032	OM_TR_INVALID_TRNAME	Invalid transaction name.
4034	OM_TR_BEGIN_FAIL	Cannot begin transaction.
4035	OM_TR_DBDOWN	Database is down at commit time.
4036	OM_TR_INVALID_TS	Failed time stamp validation
4037	OM_TR_INVALID_XID	Transaction has either incorrect or no xid
4038	OM_TR_PRIMDBDOWN	Primary database \$(dbname) is down at commit time
4039	OM_TR_REPDBDOWN	Replica database \$(dbname) is down at commit time
4041	OM_PS_NOT_FOUND	No such process
4042	OM_PS_NARGS	Too many arguments
4043	OM_PS_WAITING	Waiting by someone else
4044	OM_PS_NO_ROOT	Process has no root act
4061	OM_SHM_NOT_READY	ODA not ready
4062	OM_ODA_STACK_SIZE	Stack size mismatch
4063	OM_INVALID_OID_TABLE	Invalid object id table
4064	OM_INVALID_CLASS_TABLE	Invalid class name table
4081	OM_DB_NOT_FOUND	Not connected to this DB
4082	OM_DB_NOT_STARTED	Cannot start this DB
4083	OM_DB_NAME_INVALID	Invalid DB name or not connected

You are not using a valid database name. Database names cannot exceed 31 characters.

**For more information, on other rules related to session names, please refer to the chapter “Sessions” in the *Versant Database Fundamentals Manual*.**

4084	OM_DB_TOO_MANY_CONN	Too many connections
4085	OM_DB_CANT_DISCONNECT_PRIM	Cannot disconnect primary db

You have tried to disconnect from your session database. To use Versant, you must be in a database session. The database you use as a session workspace is the database specified in

the begin session routine. The only way to disconnect from your session database, is to end your database session.

4086	OM_DB_PERSONAL	Non-primary connection to personal database
------	----------------	---

You have tried to connect to a personal database after a session has started. You can start a session using a personal database as the session database, but after this, you can connect only to a group database and not to another personal database. The solution is to make sure that when you use a connect database routine the database that you specify as an argument is a group database and not a personal database.

4087	OM_DB_DISCONN_DIRTY	Cannot disconnect a database with dirty objects (commit or rollback before disconnecting)
4088	OM_DB_LIST_DB_NOT_FOUND	Database cannot be found in this logical database
4089	OM_DB_LIST_NOT_FOUND	Logical database was not initialized
4090	OM_DB_LIST_EXISTS	Logical database already exists
4091	OM_DB_OUT_OF_SPACE	Database out of DataVolume Space
4102	OM_LO_FE_INVALID_OBJECT	Unknown object
4109	OM_LO_BE_LOCK_BROKEN	Soft lock was broken
4110	OM_LO_BE_NO_UPGRADE	Lock cannot be upgraded
4112	OM_LO_BE_NOTFOUND	Lock not found
4113	OM_LO_RESERVED	Cannot lock comp object
4121	OM_PROF_DONE	Processing completed
4122	OM_PROF_SYNTAX	Syntax error

There is a syntax error in the application process profile that you are using for this database. You probably need to find the application process profile file and change its name. On UNIX machines, the application process file must have the same name as your database.

**For more information, about the application process file please refer to Chapter 2 “Database Profiles” on page 71.**

If problems persist, call Versant Technical Support.

4123	OM_PROF_NEGATIVE	Negative number
4124	OM_PROF_FILE	Cannot open file

4125	OM_PROF_EOF	Unexpected EOF
4126	OM_INTERNAL_ERROR	Internal OM error
4161	OM_HEAP_NOMEM	Out of front-end heap memory
4163	OM_COD_PAGE_INFO	Invalid cod page info
4164	OM_COD_ALIGNMENT	COD is aligned wrong
4165	OM_COD_ALREADY_FREE	COD is already free
4201	OM_LK_MODELEXISTS	Lock model object exists
4202	OM_LK_MODEL_SIZE	Lock model size mismatch
4203	OM_LK_MODELTABLESIZE	Lock model table size mismatch
4204	OM_LK_NOTALLOWED	Operation not allowed
4205	OM_LK_MODELNOTFOUND	Lock model object not found
4206	OM_LK_MODELNOTUNIQUE	Duplicate lock models
4208	UT_API_UNSUPPORTED	
4230	OM_SERIAL_NEG_ADD	Negative addend to serial number
4231	OM_SERIAL_OVERFLOW	Serial number overflow
4232	OM_REPLICA_MISMATCH	Replica db information mismatch
4233	OM_REPLICA_DOUBLE_FAIL	Both replica databases in a replica pair are down
4234	OM_REPLICA_WRONG_LIST	Illegal replication deferred list
4235	OM_REPLICA_NET_PARTITION	Network partition error
4236	OM_REPLICA_SYN_FAIL	Database unavailable for synchronization
4237	OM_REPLICA_SYN_DONE	Sync. to replica is done
4238	OM_REPLICA_REC_FAIL	Recursive failure during polling
4240	OM_REPLICA_UNSUPPORTED	API does not support replication
4241	OM_SYMT_TABLE_NOT_FOUND	Symbol table could not be found
4242	OM_REP_ACTICE_XA_XACT	Active external transactions present
4243	OM_REP_TR_CANT_COMMIT	Could not commit the transaction because concerned fts database \$(dbname) is down.
4260	STAT_AUTOCOLLECT_OPEN	Could not open statistics automatic collection file
4261	STAT_AUTOCOLLECT_NOT_STARTED	Statistics automatic collection is not turned on
4262	STAT_AUTOCOLLECT_ALREADY_STARTED	Statistics automatic collection is already turned on

---

4263	STAT_ERRORTABLE_NO_MATCH	Error or message not found in error table.
4264	STAT_BAD_TIME_ENV	Could not parse VERSANT_STAT_TIME environment variable
4265	STAT_BAD_STATS_ENV	Could not parse VERSANT_STAT_STATS environment variable
4266	STAT_BAD_FUNCS_ENV	Could not parseVERSANT_STAT_FUNCS environment variable
4267	STAT_FILE_WRITE	Error writing to statistics automatic collection file
4268	STAT_INVALID_STATISTIC	Invalid statistic
4269	STAT_MISSING_STATISTIC	Missing one or more statistic
4341	OM_PSR_NOMEM	VQL PARSE ERR Parser unable to allocate memory
4342	OM_PSR_NULLQUERY	Null query statement
4343	OM_PSR_SYNTAX_ERROR	VQL PARSE ERR, \$(errmsg), on/near token "\$(token)".
4344	OM_PSR_MIXED_TYPE	VQL PARSE ERR Mixed types in set of constants
4345	OM_PSR_TYPE_OVERFLOW	VQL PARSE ERR Value overflows or underflows
4346	OM_PSR_SET_PREDICATE_NOT_SUPPORTED	VQL PARSE ERR Multi-value constant in predicate is not supported in this release.
4347	OM_PSR_ATTR_NOT_FOUND_IN_DOMAIN	VQL PARSE ERR Attribute \$(attr) not in domain
4348	OM_PSR_DOMAIN_NOT_FOUND	VQL PARSE ERR Domain not found
4349	OM_PSR_NOT_AGG_TO_SCALAR	VQL PARSE ERR Cannot assign aggregate to scalar
4350	OM_PSR_NOT_AGG_OF_CHAR	VQL PARSE ERR Aggregates of char not yet supported
4351	OM_PSR_UNSUPPORTED_DBTYPE	VQL PARSE ERR Encounter Versant Type that is not yet supported
4352	OM_PSR_OBJCLASS_NOT_FOUND	VQL PARSE ERR Class for object \$(object) not found
4353	OM_PSR_ATTRCLASS_NOT_FOUND	VQL PARSE ERR Domain for attribute \$(attr) not found
4354	OM_PSR_CLASS_NOT_FOUND	VQL PARSE ERR: Class \$(class) not found.

4355	OM_PSR_TYPE_MISMATCH	VQL PARSE ERR Type mismatch on attribute \$(attr).
4356	OM_PSR_FIND_ATTR_FAIL	VQL PARSE ERR Failed to find all attribute objects for class \$(class)
4357	OM_PSR_ATTR_NOT_FOUND	VQL PARSE ERR Attribute \$(attr) not found
4358	OM_PSR_NO_VALUE_INSERT	VQL PARSE ERR: No value to insert for Attr \$(attr).
4359	OM_PSR_NULL_RESULT	VQL PARSE ERR NULL parseresult pointer is passed in
4360	OM_PSR_ATTRS_OUTBOUND	VQL PARSE ERR Number of Attributes in SELECT, UPDATE, INSERT list cannot be bigger than \$(limit)
4361	OM_PSR_PATHATTRS_OUTBOUND	VQL PARSE ERR Number of Attributes in path expression cannot be bigger than \$(limit)
4362	OM_PSR_BCD_NOT_SUPPORT	VQL PARSE ERR BCD type input value is not supported yet.
4363	OM_PSR_PARAMETER_TWICE	VQL PARSE ERR \$(index) the parameter can not be used twice in the query



## LOCKING ERRORS, SERIES - 5000

5000	OB_BADLOCK	Illegal lock mode
5001	OB_WLOCK	WLOCK is required if an object is modified
5002	OB_NULL	NULL object is not allowed
5003	OB_NOPIN	Object must be pinned

You have tried to use an invalid pointer.

Pointers are no longer valid after a transaction ends. The solution is to use a link to maintain references to a persistent object across transaction boundaries.

This error can also occur if you use a routine that unpins an object and then try to reference it with a pointer, or if you use a routine or parameter that clears that cached object descriptor table, which will invalidate both links and pointers.

For example, consider the following fragment of code in C++.

```
,,dom->beginsession( dbname, NULL ) ;
A *pa = new Persistent A(10);
pa->unpinobj();
pa->dirty();
,,dom->commit();
,,dom->endsession();
```

The error `OB_NOPIN` is going to be thrown when `dirty()` is called on the object referenced by the transient pointer `pa` after the object is unpinned from the object cache. When the `dirty()` method tries to reference the cached object descriptor table to change the status of the object, Versant finds that the object has been released from memory by the use of `unpinobj()`. In general, you will get this error only when you do something that causes a reference to the cached object descriptor table.

**For more information, on pointers and the Versant memory model, please refer to the *Versant Database Fundamentals Manual*.**

5004	OB_RESIDENT	Object cannot be deleted
5005	OB_WRONG_DB	Wrong destination db
5006	OB_NO_SUCH_OBJECT	Cannot find the object, loid = \$(loid)

You have tried to de refer a dangling or invalid link in your database.

The link you are using could be dangling because another user has deleted the object pointed to by the link.

This error will also occur if you try to use a link after clearing the cached object descriptor table with a method such as `zapcods( )`. Clearing the cached object descriptor table invalidates all links.

5009	OB_COMP_OBJ	Operation not supported on comp object
5010	OB_ANON_NOT_IN_CONT	Anonymous object not in container
5011	OB_NULL_LOID_PTR	NULL pointer to logical object id
5012	OB_INVALID_ANON	Invalid anonymous object
5013	OB_NULL_CLS_OBJ	Object has no class
5014	OB_NOT_ARCHIVED	Object was not archived from the list of input or connected data-bases
5015	OB_INVALID_OBJ_TYPE	Invalid object type
5016	OB_NO_REGION_NUMBERS	Ran out of region numbers
5017	OB_ROOT_PIN_REGION	Cannot end activity root pin region
5018	OB_BAD_PRINTABLE_LOID	Printable LOID is in bad format
5019	OB_LOCK_COUNTER_ZERO	Lock counter is zero
5020	OB_LOCK_COUNTER_MAXIMUM	Lock counter is at maximum value
5021	OB_NULL_OBJECTS	The input objects parameter is NULL
5022	OB_OBJ_NOT_REGULAR	Object is not regular
5023	OB_NO_LOCK_TO_DOWNGRADE	No locks to downgrade because of no database connection
5024	OB_DOWNGRADE_TO_NOLOCK_ONLY	Can only downgrade to no locks
5025	OB_CANT_DEL_SCHEMA_OBJ	Cannot delete class objects, attribute objects, and method objects
5026	OB_NEWOBJ	New object must be dirty
5027	OB_SINGLE_GRW_MRSW	Only one process can do group write at any time in MRSW session
5028	OB_OBJS_FROM_DIFF_DBS	All objects need to be from the same database
5029	OB_BAD_COD	Invalid COD entry

## ARCHIVE ERRORS, SERIES - 5200

5202	ARC_ARCNAME_TOOLONG	The input archive database name is too long
5203	ARC_NULL_ARCNAME	The input archive database name is null
5204	ARC_SAME_DBS	The input archive and source databases are the same
5205	ARC_NULL_OBJECTS	The input objects parameter is null
5206	ARC_BAD_ARCNAME	The input archive database is invalid, error when trying to connect to it
5207	ARC_NO_VALID_OBJS	No valid objects to archive
5208	ARC_INCOMPLETE	The archive was incomplete
5209	ARC_SYSOBJ	Trying to archive a system object
5211	ARC_OBJ_NOT_REGULAR	Trying to archive an object that is not regular
5213	ARC_CLSOBJ	Trying to archive a class object
5214	ARC_CLS_NOT_IN_DB	The object's class is not in the source database
5215	ARC_INCOMPAT_SCHEMA	Incompatible class schema between the source and archive databases
5216	ARC_NO_DBA	There is no dba associated with the source database
5217	RST_ARCNAME_TOOLONG	The input archive database name is too long
5218	RST_NULL_ARCNAME	The input archive database name is null
5219	RST_BAD_ARCNAME	The input archive database is invalid, error when trying to connect to it
5220	RST_INVALID_OBJ	Invalid object
5221	RST_NO_VALID_OBJS	No valid objects to restore
5222	RST_INCOMPLETE	The restore was incomplete
5223	RST_OBJ_NOT_IN_ARCHIVE	The object to restore was not in the archive database
5224	ARCHIVED_OBJ	The object has been archived
5225	OP_NOT_ALLOWED_ON_ARCDB	This operation is not allowed on an archive database

5226	RST_NO_TOMBSTONE_ENTRY	There's no tombstone entry for the object
5227	ARC_MORE_THAN_ONE_PDB	Cannot archive from more than one personal database to the same Archive

## QUERY ERRORS, SERIES - 5400

5400	QRY_BASE	For query processor
5401	QRY_BAD_OP	Bad comparison operator in query
5402	QRY_BAD_ATTRIBUTE	Bad attribute in query
5403	QRY_LOID_EXPECTED	Attribute type of loid expected in a (not_)isa_exact operation
5404	QRY_BAD_DOMAIN	Bad attribute to query
5405	QRY_FIRST_PATH_NOT_LOID	First attribute in path query is not a link
5406	QRY_TYPERTYPE_NOT_SPECIFIED	Keytype must be specified for path query
5407	QRY_TYPERTYPE_UNKNOWN	Invalid keytype in o_predterm
5408	QRY_NOTREACHED	Internal error
5409	QRY_ZERO_TERM_BLOCK	Predblock with no terms not supported
5410	QRY_OR_BLOCK_NOT_SINGLE	O_OR Predblock must have exact one member
5411	QRY_NO_CLASS	Class name specified in O_(NOT_)ISA_EXACT operation does not exist
5412	QRY_CLS_WITH_VSTR	Cannot have classname with O_SELECT_WITH_VSTR
5413	QRY_OBJ_WITHOUT_VSTR	Cannot have vstr_holder without O_SELECT_WITH_VSTR
5414	QRY_ATTR_WITHOUT_VSTR	Cannot have vstr_attr without O_SELECT_WITH_VSTR
5415	QRY_NO_CLS_WITHOUT_VSTR	Must have classname without O_SELECT_WITH_VSTR
5416	QRY_NO_OBJ_WITH_VSTR	Must have vstr_holder with O_SELECT_WITH_VSTR
5417	QRY_NO_ATTR_WITH_VSTR	Cannot have vstr_attr with O_SELECT_WITH_VSTR
5418	QRY_TYPERTYPE_BAD	Key type type cannot be used with this relop
5419	QRY_BAD_KEYLEN	Predicate keylen incompatible with type type
5420	QRY_STPTR_NOT_IMP	Type type O_STPTR_TYPE not implemented
5421	QRY_BAD_OPTIONS	Invalid query options

---

5422	QRY_VA_NOT_SUPPORTED	Usage of Virtual Attribute is not supported with this option
5423	QRY_OP_NOT_ALLOWED	Illegal operator-key combination
5425	QRY_CLAUSE_NOT_SUPPORTED	Unknown or Unsupported Query Clause
5440	QRY_REWRT_CANNOT_INVERT_OP	The inversion of this operator is not defined
5441	QRY_REWRT_SIGNOF_NONNUMERIC_TYPE	An attempt was made to get sign of a non-numeric value
5442	QRY_UNEXPECTED_FORMAT	Received invalid query rpc format from front end
5443	QRY_OPERAND_NOT_COLLECTION_TYPE	Operand is not of collection type
5444	QRY_UNEXPECTED_COLLECTION_TYPE	Unexpected collection type.
5445	QRY_INCOMPATIBLE_OPERANDS_TYPE	Incompatible operands type. For an operation to be evaluated successfully, its operands must be of compatible types
5446	QRY_SET_OPERATORS_NOT_SUPPORTED_ON_STRINGS	Set operators are not supported on string type operands
5447	QRY_ATTRIBUTE_NOT_FOUND	Attribute not present in class version
5448	QRY_SORT_MEMORY_EXCEEDED_QUERY_MAX	The maximum amount of sorting memory allocated to this query was exceeded
5449	QRY_SORT_MEMORY_EXCEEDED_TOTAL_MAX	The maximum amount of sorting memory allocated for all queries was exceeded
5450	QRY_PSR_SYNTAX_ERROR	VQL parser error, \$(errmsg) at \$(token).
5451	QRY_UNSUPPORTED_QUERY_TYPE	Unsupported query type
5442	QRY_UNSUPPORTED_QUERY_PROJECTION	Unsupported projection type
5453	QRY_UNSUPPORTED_QUERY_	Unsupported query class.
5454	QRY_INPUT_INVALID_LOID	Invalid loid value.
5455	QRY_VAR_ALREADY_DEFINED	Variable redefined.
5456	QRY_VARIABLE_EXPECTED	Variable expected.

---

5457	QRY_UNSUPPORTED_ORDERBY _EXPRESSION	Unsupported order by expression.
5458	QRY_BAD_COLLECTION	Invalid or bad collection.
5459	QRY_BAD_CLASS_NAME	Invalid or bad class name.
5460	QRY_ATTR_TYPE_LOID	Attribute type loid expected.
5461	QRY_INVALID_TYPE	Not a valid data type.
5462	QRY_INVALID_COLLECTION_ TYPE	Not a valid collection type.
5463	QRY_COLLECTION_TYPE _UNSUPPORTED	Unsupported collection type.
5464	QRY_ATTR_NOT_FOUND	Attribute does not exist for class.
5465	QRY_INVALID_PARSE_BLOCK	Invalid parse block.
5466	QRY_PREDICATE_UNEXPECTED_TYPE	Unexpected query predicate type
5467	QRY_PREDTERM_UNEXPECTED_TYPE	Unexpected query predterm type
5468	QRY_UNSUPPORTED_COND_OP	Unsupported conditional operator in query
5469	QRY_PARAM_NOT_SUBSTITUTED	Query parameter \$(paramName) not substituted
5470	QRY_CAN_NOT_TAKE_CANDIDATE _COLLECTION	Candidate collection can not be set for select with vstr or can- didate collection queries
5471	QRY_UNSUPPORTED_DATA_TYPE	This data type is not supported in queries
5472	QRY_INVALID_DATA_TYPE	This data type is invalid
5473	QRY_PARAM_SIZE_MISMATCH	Given param size does not match with size of parameter's data type
5474	QRY_PARAM_STRING_NOT_NULL _TERMINATED	Given param string is not null terminated
5475	QRY_PARAM_STRING_SIZE_MISMATCH	Given param string length does not match with specified param Size
5476	QRY_PARAM_NOT_FOUND	Param \$(paramName) not found
5477	QRY_CANDIDATE_NOT_FOUND	Candidate \$(candidateName) not found
5478	QRY_API_NULL_ARGUMENT	Api argument \$(argnum) is NULL
5479	QRY_API_INVALID_START_POS	Invalid start position

---

5480	QRY_UNSUPPORTED_COND_OP_IN _COMPLEXEXPR	Conditional operator \$(condop) in complex expression not yet supported
5481	QRY_FETCH_SIZE_EXCEEDED_ MAX	Fetch size exceeded allowed maximum value
5482	QRY_MAX_RESULT_COUNT_EXCEEDED _MAX	Max result count exceeded allowed maximum value
5483	QRY_INVALID_HANDLE	Query handle is invalid
5484	QRY_INCORRECT_SESSION	This query does not belong to this session
5485	QRY_RESULT_UNUSABLE	Query result is unusable and should be destroyed
5486	QRY_RESULT_INVALID_HANDLE	Query result handle is invalid
5487	QRY_RESULT_INCORRECT_SESSION	This query result does not belong to this session
5488	QRY_RESULT_BACKWARD_SEEK_NOT _SUPPORTED	Moving backward in query result is not supported
5489	QRY_RESULT_RECORDNUM_EXCEEDED _MAX	Recordnum exceeded allowed maximum value
5490	QRY_RESULT_CURRENT_POSITION_AT _BEFORE_FIRST	Current cursor position is at before the first record
5491	QRY_RESULT_ENCOUNTERD_END	Encountered end of query result
5492	QRY_RESULT_INVALID_ PROJNUM	Invalid projnum
5493	QRY_RESULT_INVALID_RECORDNUM	Invalid record number
5494	QRY_NULL_DOMAIN_ATTRIBUTE	Null domain attribute in path expression
5500	CSR_BASE	For cursor processor
5501	CSR_NOEXIST	The specified cursor is not currently open
5502	CSR_BAD_FLAGS	Reserved values not zero or invalid options
5503	CSR_POS_UNDEF	Severe error occurred on previous cursor operation
5504	CSR_RELS_CURSOR	Cannot release cursor
5505	CSR_BAD_OPTIONS	O_CSR_DROP_RLOCK bit is not allowed to be set in the options parameter



---

## VERSION ERRORS, SERIES - 5600

5601	VSN_CANT_DELE_RELEASE	Cannot delete released version
5602	VSN_NO_CLASSOBJ	Class object is not allowed
5603	VSN_BAD_STATUS	Bad version status
5604	VSN_NO_SUCH_VERSION	Version does not exist
5605	VSN_CLASS_NOT_COMP	Class of version object not compatible
5606	VSN_BAD_CLASSNAME	Bad class name
5607	VSN_CANT_UPD	Cannot update working or released version
5608	VSN_NON_LEAF	Non-leaf version cannot be transient
5610	VSN_NO_CYCLIC_VDG	Cyclic VDG is not allowed
5611	VSN_CANT_VSN_COMPOBJ	Cannot version comp object
5612	VSN_BAD_INDEX	Bad index into pinned objects
5613	VSN_BAD_DEFAULT	No default version
5614	VSN_BAD_VSNNO	No version number
5615	VSN_BAD_NODE	No node in VDG for version
5616	VSN_BAD_INPUT	Bad input arguments
5617	VSN_DELETED	Version has been deleted

## SCHEMA ERRORS, SERIES - 6000

6000	SCH_BASE	For schema manager
6001	SCH_CLASS_DEFINED	Class already defined
6002	SCH_CLASS_UNDEFINED	Class \$(class) undefined
6003	SCH_NULL_CLSNAME	Null class name
6004	SCH_BAD_SUPERCLASS	Bad superclass \$(class)
6005	SCH_BAD_ATTRNAME	Bad attribute name \$(attribute)
6006	SCH_BAD_METHNAME	Bad method name
6007	SCH_BAD_DOMAIN	Bad domain \$(class)
6008	SCH_BAD_BUFFER	Bad return buffer
6009	SCH_NOMEM	Out of schema memory
6010	SCH_NON_LEAF_CLASS	Update to non-leaf class
6011	SCH_BAD_CACHE	Bad object cache
6012	SCH_BAD_CLSOBJ	Bad class object on disk
6013	SCH_CREATE_CLSOBJ	Cannot create class object
6014	SCH_NULL_DBNAME	Null database name
6015	SCH_NULL_ATTRNAME	Null attribute name
6016	SCH_NULL_METHNAME	Null method name
6017	SCH_BAD_ATTROBJ	Bad attribute object
6018	SCH_BAD_METHOBJ	Bad method object
6019	SCH_CANT_CACHE_GDBCLS	Cannot cache global class
6020	SCH_CANT_UPD_CLS	Cannot update class
6021	SCH_NO_SUCH_INDEX	Non-existing index
6022	SCH_NON_INDEXABLE	Specified index cannot be defined on the given attribute
6023	SCH_NULL_OBJECT	A NULL object
6024	SCH_DBNAME_NOTMATCH	Database name does not match
6025	SCH_CANT_MOVE_VSNOBJ	Cannot migrate version
6026	SCH_CANT_MOVE_CLSOBJ	Cannot migrate class
6028	SCH_CANT_DROP_INHR_ATTR	Cannot drop inherited attribute
6029	SCH_CANT_DROP_INHR_METH	Cannot drop inherited method
6030	SCH_BAD_CLS_STRUCT	Bad class structure
6031	SCH_CLSNAME_TOOLONG	Class name too long
6032	SCH_ATTRNAME_TOOLONG	Attribute name too long
6033	SCH_METHNAME_TOOLONG	Method name too long
6035	SCH_CANT_UPD_SYCLS	Cannot update system class

---

6036	SCH_CANT_INDEX_SYSCLS	Cannot index on system class
6037	SCH_CANT_UPD_SYSOBJ	Cannot update system object
6038	SCH_CANT_CREATE_SYSOBJ	Cannot create system object
6039	SCH_CANT_MOVE_SYSOBJ	Cannot move system object
6040	SCH_CANT_UPD_INHRDOMAIN	Cannot update inherited domain
6041	SCH_CANT_UPD_DOMAIN	Cannot update non-null domain
6042	SCH_BAD_CLS_TBL	Bad front-end class cache
6043	SCH_BAD_SCH_CACHE	Bad back-end schema cache
6044	SCH_CANT_INHERIT_SELF	Cannot inherit from itself
6045	SCH_OBJ_NOT_FOUND	Cannot find object
6046	SCH_CLSDEF_CONFLICT	Conflicting class definitions in databases
6047	SCH_CANT_EXPAND_MEM	Cannot expand memory - internal only
6049	SCH_BAD_ATTR_OFF	Bad attribute offset
6050	SCH_BAD_ATTR_POS	Bad attribute position
6051	SCH_CANT_MOVE_COMP	Cannot migrate composite
6052	SCH_REF_TO_ANON	Ref to anonymous object
6053	SCH_DUP_ATTRNAME	Duplicate attribute name
6054	SCH_DUP_METHNAME	Duplicate method name
6055	SCH_BAD_INDEXTYPE	Index type must be O_IT_BTREE or O_IT_HASH
6056	SCH_OBJ_SIZE_MISMATCH	Mismatch between bufdesc length and object size in the database
6057	SCH_NO_CLASS_CLASS	Cannot find storage of class class
6058	SCH_GENERATE_RESOBJ_FAILED	Failed to generate resident schema objects
6059	SCH_CLS_INIT_FAILED	Failed to init schema class variables
6060	SCH_RELEASE_RESOBJ_FAILED	Failed to release resident objects or class table
6061	SCH_CRT_RESOBJ_FAILED	Failed to create resident schema objects
6062	SCH_NULL_OBJECTS	The input objects parameter is NULL
6063	SCH_MIGR_NO_VALID_OBJS	No valid objects to migrate

6064	SCH_CLASS_NOT_IN_SOURCE_DB	The class of the object to migrate is not in the source database
6065	SCH_OBJ_NOT_REGULAR	Object is not regular.
6066	SCH_CANT_RENAME_INHR_ATTR	Cannot rename an inherited attribute.
6067	SCH_OBJ_HAS_NO_ATTR	Class instance has no attributes
6068	SCH_CLSDEF_CHANGED	Class definition has changed
6069	SCH_CANT_EVOLVE_IF_DBDOWN	Schema evolution is not allowed if any connected replica database is down.
6070	SCH_WRONG_SIZE_VSTR	The input vstr size is incorrect
6071	SCH_LAYOUT_ATTR_NAME	Bad attribute name in o_attrlayout
6072	SCH_LAYOUT_ATTR_TYPE	Bad attribute type in o_attrlayout
6073	SCH_INDEX_CONFLICT	A similar index exists on attribute
6074	SCH_CLS_MUST_BE_EMPTY	Class must be empty to be clustered
6075	SCH_CANT_DROP_SYSTEM_CLASS_ATTR	Cannot drop attribute belonging to a system class
6076	SCH_CANT_DROP_SYSTEM_CLASS	Cannot drop system class
6077	SCH_CANT_MODIFY_SYSTEM_CLASS	Cannot modify system class
6078	SCH_CANT_RENAME_SYSTEM_CLASS_ATTR	Cannot rename system class attribute
6079	SCH_CANT_RENAME_SYSTEM_CLASS	Cannot rename system class

## CONTAINER ERRORS, SERIES - 6200

6200	FS_BASE	For fasload/fasdump
6203	FSD_BAD_ATTROBJ	Bad attribute object detected in fasdump
6204	FSL_BAD_ATTROBJ	Bad attribute object detected in fasload
6205	FSD_CANT_DUMP_REMOTE OBJ	Cannot dump remote object
6206	FSL_CANT_LOAD_REMOTE OBJ	Cannot load remote object
6207	FSD_CANT_DUMP_LOGGED	Cannot dump with logging on
6208	FSL_OBJ_NOT_FOUND	Cannot find object to load
6209	FSL_TOO_MANY_DBIDS	Too many different dbids in LOIDs
6300	CONT_BASE	
6302	CONT_NULL_DBNAME	Db name is null
6303	CONT_OBJ_NOT_CONTAINER	Not a container object
6304	CONT_NULL_CLS_OBJECT	Null class object
6305	CONT_CANT_CREATE_SYSOBJ	Class object is resident class
6306	CONT_NULL_OBJECT	Null object
6307	CONT_NOT_ANON_OBJ	Cannot be an anonymous object
6308	CONT_CANNOT_INSERT_AT	Cannot insert container object into associative table
6309	CONT_BE_NO_OBJ_TO_DELETE	No object to delete
6310	CONT_MEM_NO_MEM	Out of mem writing cont
6311	CONT_OBJ_INVALID_OBJ	Bad object type passed
6312	CONT_CANNOT_EMBED_CONTAINERS	Container in container not allowed
6313	CONT_READ_NO_MEM	Out of mem reading container
6314	CONT_ROOT_NO_MEM	Out of mem read container object
6315	CONT_ROOT_BAD_TYPE	Bad object type in container
6316	CONT_SCH_OBJ_NOT_PINNED	Object not pinned
6317	CONT_SCH_NOT_ANON_OBJ	Trying to upgrade non-anonymous object
6318	CONT_VAR_NO_MEM	Out of memory in variable size attribute
6319	CONT_CANNOT_REF_OWN_CONTAINER	Ref to own container object
6320	CONT_WRITE_NO_DISK_SPACE	Out of disk space
6322	CONT_MEM_NULL_DB	Bad db
6323	CONT_VAR_REF_TO_ANON	Ref to anonymous object

6324	CONT_SCH_NULL_ANON	Ref to ext anon
6325	CONT_VAR_REF_DEL_OBJ	Ref to deleted object
6326	CONT_WRITE_REF_DEL_OBJ	Ref to deleted object
6327	CONT_WRITE_BAD_TYPE	Unexpected object type
6328	CONT_NULL_PARAM	Null pointer was passed in
6329	CONT_SE_TOO_MANY_CONTAINERS	Too many container in session
6330	CONT_OBJ_SCH_OBJ	Cannot put schema object in container
6333	CONT_MEM_ROOT_IS_PINNED	Root is still pinned

---

## EVENT NOTIFICATION ERRORS, SERIES - 6500

6500	EV_BASE	
6501	EV_BAD_USER_DATA	Invalid user input data
6502	EV_NULL_OID	NULL object logic id
6503	EV_BUFFER_TOO_SMALL	Buffer too small
6504	EV_BAD_BUFFER_PTR	Bad buffer pointer
6505	EV_AUX_INFO_MISMATCH	Aux info length mismatch
6506	EV_WRONG_EVENT_ORDER	Invalid event
6507	EV_INVALID_KEY	Invalid database path name
6508	EV_IPC_NOT_EXIST	
6509	EV_IPC_REMOVED	IPC queue removed
6510	EV_BAD_IPC_ID	Invalid IPC id
6511	EV_CANT_START_DD	Cannot start event daemon
6512	EV_EVENT_LOST	An event not delivered
6513	EV_OP_NOT_ALLOWED	Not a front-end operation
6514	EV_PROCESS_INIT	Cannot initialize daemon
6515	EV_BAD_HANDLE	Bad event queue handle
6516	EV_NOT_ACTIVATED	Event notification disabled
6517	EV_INVALID_REG_ID	Invalid registration id
6518	EV_NOT_REG_OWNER	Not owner of registration
6519	EV_BAD_EVENT_PAIR	Lower event greater than higher event
6520	EV_BAD_DEF_FLAGS	
6521	EV_BAD_EVENT_TYPE	Not a proper event type
6522	EV_BAD_OPTION	Invalid option
6523	EV_BAD_FLAGS	Invalid flags
6524	EV_DAEMON_NOT_RUN	Event daemon not running
6525	EV_NULL_EVENT	NULL events
6526	EV_OBJ_NOT_EXIST	Object \$(loid) does not exist
6527	EV_IPC_VIOLATED	Permission denied to access message queue
6528	EV_NO_MSG	No message
6529	EV_NO_REG	No event registrations
6530	EV_STILL_ACTIVATED	Event notification activated

6531	EV_NOT_INITIALIZED	ev in memory structure has not been initialized. (try calling o_initen(O_EV_ENABLE),and o_xact if in persistent registration mode)
6532	EV_PTR_CAN_NOT_NULL	User input pointer should not be null
6533	EV_CLASS_INVALID_NUM_ OBJ	class %s has %d num instances, it should only have %d
6534	EV_BAD_DB	Inconsistent pewcrsistent ev schema (possibly caused by user changing its schema)
6535	EV_ASSERT_FAIL	Internal assertion failure
6536	EV_REG_ALL_OBJ_DEL	ev registration \$(loid) is invalidated, all of its registered objects have been deleted
6537	EV_ERR_INCOMPATIBLE	Pre 5.0 app using event with new event mode is not supported
6538	EV_FE_DIED	Frontend has died
6539	EV_ERR_DB_SHUTDOWN	Db is shutting down. (possible by stopdb -st
6540	EV_ERR_DAEMON_EXIST	A event daemon exists. Mutliple event daemon is not supported in current release



## VIRTUAL ATTRIBUTE ERRORS, SERIES - 6600

6600	VA_BASE	
6601	VA_PARSE_FAILED	Invalid syntax of VA
6602	VA_NOT_INITIALIZED	VA is not initialized
6603	VA_VAT_NOT_FOUND	VAT is not defined
6604	VA_VA_INIT_ERR	VA initialization error
6605	VA_KEY_NOT_MATCH	Key items doesn't match attribute items
6606	VA_KEY_PARSE_FAILED	Key has wrong format
6607	VA_TYPE_UNSUPPORTED	Wrong Type Or Type Is Not Supported Yet
6608	VA_KEY_OVERFLOW	Key Value Overflowed
6609	VA_NOT_VALID_CMP_FLAG	Invalid Compare Method
6610	VA_INTERNAL_ERROR	Va Internal Error
6611	VA_EMPTY_KEY	Key Is An Empty String
6612	VA_HASH_INDEX_NOT_SUPPORTED	Hash Index On VA Not Supported
6613	VA_INDEX_ALREADY EXISTS	duplicate index not allowed
6614	VA_NO_SUCH_INDEX	Non-existing index
6615	VA_INDEX_NOT_SUPPORTED_IN_FTS	Index operations on replica databases not supported yet
6616	VA_EXISTED_IN_HASHTABLE	Entry Is In Hashtable Already
6617	VA_NO_CLASS_VERSION_FOUND	Class Doesn't Have Any Instance
6618	VA_NOT_INDEXABLE	Indexing Not Supported On This Virtual Attribute
6619	VA_INSUF_ARGS	Insufficient No. Of Arguments
6620	VA_MEMALLOC_ERROR	Memory Allocation Error
6621	VA_INCORRECT_TYPE	Received Incorrect Type Of Argument
6622	VA_NULL_SPECIFIC	Va_Getspecific Returned NULL
6623	VAT_CONSTRUCTOR_ERROR	Error \$(Err) In Vat_Constructor
6624	VAT_DESTRUCTOR_ERROR	Error \$(Err) In Vat_Destructor
6625	VAT_VALUES_ERROR	Error \$(Err) In Vat_Values
6626	VAT_COMPARE_ERROR	Error \$(Err) In Vat_Compare
6627	VA_QUERY_OPERATOR_NOT_SUPPORTED	Operator Not Supported

## SYSTEM UTILITY ERRORS, SERIES – 7000, 9000, 11000, 12300

7000	UT_OK	No error
7001	UT_DB_NOT_FOUND	DB directory not found

A database directory was not found, probably because the `VERSANT_DB` configuration parameter does not point to a database root directory that has a subdirectory with the same name as the database you want to use. Check to make sure that there is a directory with the same name as the database under the directory returned by invoking the environment utility as `oscp -d`.

**For more information, on database directories, `VERSANT_DB`, refer to “Directories and Files” on page 281 in “Chapter 5 - Directories and Files”.**

**For more information on `oscp`, refer to “oscp” on page 203 in “Chapter 4 - Database Utilities”.**

7002	UT_DB_NOT_PERSONAL	Not a personal DB
7003	UT_DB_NO_ACCESS	No access rights to DB dir

You do not have appropriate permissions on the directory containing the database you are trying to access. The solution is to check permissions on the database directory and change them as appropriate.

7004	UT_BE_PROFILE_ERROR	Error parsing back-end profile
7005	UT_FE_PROFILE_ERROR	Error parsing front-end profile
7006	UT_ER_NOFILE	Cannot find error message file error.msg
7007	UT_ER_NO_MSG	Cannot find error message

The database server process needs to report an error, but it cannot find the error message file.

This problem can occur if the server process is on a different machine and is running a different Versant release than the application process. The solution is to assert the release number on the server using the `VERSANT_REL@server` configuration parameter on the client machine. In any case, the server machine must be running a release of Versant that is numerically equal or higher to the release running on the client machine.

This problem can also occur on a UNIX machine if the file `/etc/.oscxyyzz` on the server machine does not have the correct entry for `VERSANT_ROOT`. In this case, you can either correct the `VERSANT_ROOT` entry in the `.oscxyyzz` file on the server machine or set the environment parameter `VERSANT_ROOT@server` on the client machine to point to the Versant software root directory.

Another possible reason for this error is that the error messages file `error.msg` has not been installed on the server machine, which could occur if a vendor has supplied you with a run-time version of Versant.

**For more information about the `VERSANT_REL@server` and `VERSANT_ROOT@server` configuration parameters, refer to Chapter 5 “Configuration Parameters” on page 315.**

7008	UT_DB_ALREADY_STARTED	Database already started in another machine
7009	UT_WRONG_OBE_VERSION	OBE version doesn't match client's
7010	UT_UNKNOWN_UTILITY	Unknown database utility
7011	UT_DB_EXISTS	Database already exists
7012	UT_INVALID_PARAMETER	Invalid parameter
7013	UT_ER_CREATE_DBDIR	Failed to create database directory
7014	UT_ER_CHGOWNER_DBDIR	Failed to change owner of database directory
7015	UT_ER_CHGGROUP_DBDIR	Failed to change group of database directory
7016	UT_ER_CREATE_BEPROFILE	Failed to create server process profile. Check disk space, permissions etc.
7017	UT_ER_CHGOWNER_BEPROFILE	Failed to create default database profiles
7018	UT_ER_CREATE_FEDIR	Could not create application process profile. Check disk space, permissions, etc.
7019	UT_ER_CHGOWNER_FEPROFILE	Failed to change owner of FE profile
7020	UT_ER_CHGGROUP_BEPROFILE	Failed to change group of database BE profile
7021	UT_ER_DBDIR_NOACCESS	Database directory not found or no access permission

You may have tried to create a database with the `createdb` utility without first creating the database directory and database support files. In this case, the solution is to run the `makedb` utility before running `createdb`.

**For more information refer to “Database Creation Basics” on page 54, in “Chapter 2 - Database Creation”.**

Alternately, your login name may have a space in it, although Windows allows spaces in login names, Versant does not.

7022	UT_ER_SYSTEMVOL_NOACCESS	No access to system volume
7023	UT_ER_SYSTEMVOL_FILE_TYPE	System volume already exists and is not a raw device
7024	UT_ER_SYSTEMVOL_CREATE	Unable to create system volume
7025	UT_ER_PHYSICALLOG_NOACCESS	No access to physical log volume
7026	UT_ER_PHYSICALLOG_EXISTS	Physical log volume already exists and is not a raw device
7027	UT_ER_PHYSICALLOG_CREATE	Unable to create physical log volume
7028	UT_ER_LOGICALLOG_NOACCESS	No access to logical log volume
7029	UT_UR_LOGICALLOG_FILE_TYPE	Logical log volume already exists and is not a raw device
7030	UT_ER_LOGICALLOG_CREATE	Unable to create logical log volume
7031	UT_ER_DBID_NOACCESS	Failed to access database ID file
7032	UT_ER_FORMAT_LOGFILES	Failed to format log files
7033	UT_ER_FORMAT_DATABASE	Failed to format database
7034	UT_ER_SDA_INIT	Failed to initialize shared memory

Shared memory is not configured on this system.

The solution is to configure shared memory on the system according to the instructions in the installation notes.

For example, on Sun machines, the system configuration file is in `/usr/kvm/sys/sun4/conf`. The name of the file depends on the system administrator who has configured your system. The following two lines need to be present in this file for shared memory to be available in your system,

```
options IPCSHMEM
```

`options SHMMNI=nnn`

In the above, `nnn` represents the maximum number of shared memory segments that can be made available in your system.

Please set `SHMMNI` to the value suggested by Versant in the "Platform Notes" chapter of the Release Notes.

7035	UT_ER_VOLUME_SIZE	Invalid Volume Size
7036	UT_ER_STARTDB	Failed To Start Database
7037	UT_ER_VOLUME_OFFSET	Invalid Volume Offset
7038	UT_ER_VOLUME_NOACCESS	No Access To Volume
7039	UT_ER_VOLUME_EXISTS	Volume Already Exists
7040	UT_ER_VOLUME_CREATE	Unable To Create Volume
7041	UT_ER_VOLUME_FORMAT	Failed To Format Volume
7042	UT_ER_VOLUME_DELETE	Failed To Delete Volume
7043	UT_ER_BEGIN_SESSION	Failed To Begin Session
7044	UT_ER_UNAUTHORIZED_ACCESS	Unauthorized Access
7045	UT_ER_DB_INUSE	Database Is In Use
7046	UT_ER_DB_INCONSISTENT	Database May Be Inconsistent
7047	UT_ER_USER_NOTFOUND	Database User Not Found
7048	UT_ER_USER_FOUND	Database User Found
7049	UT_ER_USER_IS_DBA	User Is DBA
7050	UT_ER_ROLLBACK	Failed To Rollback Database
7051	UT_ER_SDA_CRASHED	Shared Memory Crashed

The database server process has died due to lack of system resources and/or due to an inconsistency seen by the Versant database.

If this occurs, check the file `LOGFILE` in your database directory, look at error messages reported there, and execute the solution as per the recommendation Or call Versant Technical Support.

For example, in `LOGFILE` you might see the following error message,

0964, `SL_ATTACH_EXTEND_SEG` Cannot attach extension segments in the `LOGFILE`.

In this case, error 0964 is an error for which you have a solution (presented above) and you can execute the solution as per the recommendation.

7052	UT_ER_SDA_GRPSHARE	Unable To Share SDA For Group
7053	UT_ER_PROCESS_INIT	Process Initialization Failed
7054	UT_ER_RECOVERY	Database Recovery Failed

---

7055	UT_ER_CLEANBE	Failed To Start Cleanbe Process
7056	UT_ER_HEAP_NOMEM	No More Heap Space
7057	UT_ER_MOUNT_TAPE	Tape Not Mounted
7058	UT_ER_FE_CLEANUP	Failed To Clean Up Front Resources
7059	UT_ER_UNKNOWN_DB	Unknown Database
7060	UT_ER_LOCKMODEL_FILE	Cannot Get New Lock Model From File
7061	UT_ER_DB_LOCKED	Database Already Started
7062	UT_ER_DETACH_ODA	Failed To Detach From ODA
7063	UT_ER_PARSE_BEPROFILE	Error %D: Parsing Back-End Profile At %S
7064	UT_ER_NOT_DB_OWNER	Not Database Owner
7065	UT_ER_NOT_GROUP_MEMBER	Not A Group Member
7066	UT_ER_DBID_CREATE	Cannot Create Database ID File At %S
7067	UT_ER_DBID_LOCK	Error %D: Obtaining Lock On DBID File
7068	UT_ER_MAKE_DB_DIRECTORY	Failed To Make Db Directory
7069	UT_ER_REMOTE_ACCESS	Failed To Invoke Utility Remotely

There has been a network error or a login error.

If the problem is a login error and the error occurred on a personal computer, it may mean that the configuration parameter `VERSANT_USER` has not been set to a valid user of the database involved. In this case, set `VERSANT_USER` to a valid user name or have the DBA for the database add the current user name to the list of authorized user.

This error may also mean that your user name is not a valid login name on the remote machine. In this case, have the system administrator for the remote machine to add your user name to the list of valid login names on the machine where the database is located.

This error can occur if the remote `inetd` daemon is not able to start the `ss.d` process. In this case, the solution is to update the `/etc/inetd.conf` file with the correct entry and then restart the `inetd` daemon. For further information, see the installation instructions in the Release Notes.

This error can also occur if the `ss.d` process is not able to start the server process for the database. This can occur in the following situations,

- if the Versant machine configuration file `/etc/.oscxyynn` (where `xyynn` corresponds to your release number) does not have the right setting for `VERSANT_ROOT` (which specifies the software root directory).
- if different releases are running on the client and server and the configuration parameter `VERSANT_REL@node` has not been properly set on the client machine, or
- if the number of the Versant release running on the server is not numerically equal to or greater than the client release number.

This error can also occur if the database server process crashes while the application process was communicating with it, in which case there will be a core dump. This could happen due to resource limitations such as lack of shared memory or semaphores. In this case, the cause is normally written to a file called `LOGFILE` under the Versant database directory. If `LOGFILE` does not contain the information you need to solve the problem, please call Versant Technical Support.

**For more information on the `.oscxyynn` machine configuration file, the `VERSANT_ROOT` and `VERSANT_REL` configuration parameters, and the `LOGFILE` error file, please refer to the *Versant Database Administration Manual*.**

7070	UT_ER_INTERNAL	Internal Error
7071	UT_ER_READ_BE_PROFILE	Cannot Read Back-End Profile
7072	UT_ER_WRITE_BE_PROFILE	Cannot Write Back-End Profile
7073	UT_ER_READ_FE_PROFILE	Cannot Read Front-End Profile
7074	UT_ER_WRITE_FE_PROFILE	Cannot Write Front-End Profile
7075	UT_ER_EXCEED_MAXVOLUMES	Exceed Maximum Number Of Volumes
7076	UT_ER_OSC_DBID_EXISTS	Osc-Dbid File Already Exists
7077	UT_ER_BAD_GEN_OP	Invalid Operation For Ut_Genericrpc
7078	UT_ER_BAD_GEN_REP_MODEL	Invalid Replicamodel For Rep_Genericrpc
7079	UT_SS_INVALID_PORTRANGE	Invalid Port Range Specified (Valid Range 1 Thru 32767)
7080	UT_SS_UNKNOWN_CLIENT	Unable To Get Peer Name Of Cli- ent. %M
7081	UT_SS_LOGIN_ERROR	Error Receiving Login Message From %S. %M
7082	UT_SS_SEND_ERROR	Unable To Send Response To %S. Errno, %M
7083	UT_SS_UNAUTHORIZED_ACCESS	Unauthorized Remote Login, %D

This error probably means that you are logged on as `ROOT`. The solution is to not use Versant as a superuser. Versant checks for superuser status and raises this error for security reasons.

7084	UT_SS_UNKNOWN_PATH	Unable To Obtain Oscpath To %S. Errno, %M
7085	UT_SS_CONNECTION_TIMEOUT	Connection From %S Has Been Timed Out.
7086	UT_SS_CONNECTION_DROPPED	Connection From %S Has Been Dropped.
7087	UT_SS_INCOMPATIBLE_RELEASE	Fail To Find A Compatible Release
7088	UT_ER_WRONG_OSC_PATH	Wrong Osc Path %S
7089	UT_ER_OSC_PATH_NOT_FOUND	Osc Path %S Not Found

A remote machine cannot find the location of the server process executable.

If the remote machine is a UNIX machine running the same Versant release as your machine, confirm that the `VERSANT_ROOT` configuration parameter in the file `/etc/.oscxyyzz` on the remote machine points to the correct Versant software root directory.

If the remote machine is running a different Versant release than your machine, confirm that the release number on the remote machine is numerically higher than the release number on your machine. If the server release number is higher than the application release number, set the `VERSANT_ROOT@server` configuration parameter as an environment variable on the application machine to point to the Versant software root directory on the remote machine.

**For more information, on the `/etc/.oscxyyzz` file and the `VERSANT_ROOT@server` configuration parameter, refer to Chapter 5 “Configuration Parameters” on page 315.**

7090	UT_SS_BAD_BUFSIZE	Network buffer size is too large
7091	UT_INCOMPATIBLE_LOCKFILE	Incompatible .lock file
7092	UT_DB_IS_UNSTARTABLE	Database is unstartable
7093	UT_LOCKFILE_NOT_FOUND	.lock file not found
7094	UT_ER_ADD_USR_TO_PDB	Cannot add users to personal databases
7095	UT_ALIGN_NOT_POWER_OF_2	O_align not on a power of 2
7096	UT_ER_DBDIR_EXISTS	Database directory already exists
7097	UT_ER_BEPROF_EXISTS	Back end profile file already exists



---

7098	UT_ER_FEPROF_EXISTS	Warning: using an existing front end profile
7099	UT_ER_CLEANFE	Cleanfe terminated abruptly
7100	UT_ER_UNKNOWN_LEVEL	Unknown level
7101	UT_ER_NEED_LEVEL_0	Restore level 0 first
7102	UT_ER_NEED_LEVEL_1	Restore level 1 first
7103	UT_ER_WRONG_DBID	Wrong dbid
7104	UT_ER_NO_DBID	Cannot create dbid
7105	UT_ER_NO_DIRECTORY	A database directory is required. Create one with makedb or vutil.
7106	UT_ER_BACKING_UP	Another process is backing up already
7107	UT_UNCONVERTED_AND_STARTED	Convertddb requires that the database be stopped
7108	UT_DIFFERENT_HOSTS	Both databases must be on the same machine -- "\$(db1)" and "\$(db2)" are not.
7109	UT_LOCAL_ONLY	Remote databases are unsupported. Try running on "\$(host)".
7110	UT_TRIAL_EXPIRED	Trial period expired on evaluation copy
7111	UT_DBID_EXCEED_LIMIT	Dbid can not exceed ut_dbid_user_max (0xff7f)
7112	UT_ER_VOLUME_DUPENTRY	Database profile.be has duplicate entry for data volume
7113	UT_ER_VOLUME_NEEDADD	Need run 'addvol' for the data volume entry in profile.be
7114	UT_ER_RESTORE_PREV_VER	Restore from backup of previous version not allowed
7115	UT_ER_PASSWDS_DONOTMATCH	The passwords do not match!!!
7116	UT_ER_2PROCESS	Must be a two process connection.
7117	UT_ER_SOME_OBJECTS_DIFFERENT	Some objects are different in two databases
7118	UT_DB_IS_UNRESTORED_HABACKUP	The database device is an unrestored habackup
7119	UT_ER_DBALOGGING_LEVEL_IS_ZERO	The DBA logging level is zero, so DBA logging is disallowed
7120	UT_ER_DBID_EXISTS	Database id already exists in osc-dbid file

---

7121	UT_ER_DB_ENTRY_EXISTS	Database entry already exists in osc-dbid file
7122	TARGET_DB_NOT_CREATED	Vmovedb requires that target database be created
7123	UT_FAILED_TO_MOVE_ALL_OBJECTS	Could not move all objects
7124	UT_NO_OBJECTS_TO_MOVE	No objects to move
7125	UT_NO_REPLICA_MOVE	Vmovedb is not allowed on a replica pair of databases
7126	UT_VMOVEDB_INVALID_TARGET	The specified database is invalid for vmovedb
7127	UT_VBACKUP_INCOMPLETE	Vbackup did not backup all pages
7128	UT_ER_MAKEDB_PROFILE_NOTFOUND	Makedb did not find the profile file at the Specified location
7129	UT_ER_ADDVOL_PROFILE_UPDATION_FAILED	Addvol failed to update server profile file with datavol entry
7130	UT_ER_ADDVOL_DATAVOL_ENTRY_ALREADY_EXISTS	Datavol entry already exists in server profile file
7131	UT_ER_DATAVOL_ALREADY_EXISTS	Data volume at the specified location already exists
7132	UT_ER_VBACKUP_RESTORE_REMOVE_DATABASE	Vbackup restore process is unable to remove creation time database files of renamed database
7133	UT_ER_VBACKUP_DUPLICATE_RESTORE	Another process is doing restore already
7134	UT_ER_USER_ABORT_REQUEST	Aborted at user's request.
7135	UT_ER_ALREADY_CONVERTED	Database version is same as current release version. No need to convert.
7136	UT_ER_STARTUP_ABSOLUTE_PATH	Error launching startup executable. Absolute path is not allowed.
7137	UT_ER_STARTUP_IS_DIR	Error launching startup executable. \$(file) is a directory.
7138	UT_ER_STARTUP_NOT_FOUND	OS err #\$(oserr) while launching startup executable. \$(file) could not be found.
7139	UT_ER_STARTUP_INCORRECT_PERMISSIONS	Error launching startup executable. \$(file) has write permission for users other than DBA.

---

7140	UT_ER_STARTUP_IS_LINK	Error launching startup executable. \$(file) is a symbolic link.
7141	UT_ER_STARTUP_WRITE_LOG	OS err #\$(oserr) while writing to log file. Launching startup executable failed.
7142	UT_ER_STARTUP_LAUNCH	Launching startup executable \$(file) failed.
7143	UT_ER_STARTUP_SET_PROCID	Error while trying to save the startup executable process-id
7144	UT_ER_STARTUP_ALREADY_RUNNING	Cannot start database. A previous instance of the startup script is already running with process-id \$(pid)
7145	UT_ER_STARTUP_GET_DBDIR	Error retrieving database directory.
7146	UT_ER_STARTUP_OPEN_LOG	Error while opening \$(file) for writing.
7147	UT_ER_STARTUP_NOT_EXECUTABLE	Error launching startup executable. \$(file) does not have execute permission for DBA.
7180	LIC_ERR_ARGS	Error in arguments passed
7181	LIC_ERR_NODE	Component not licensed
7182	LIC_ERR_DATE	License expired
7183	LIC_ERR_ENVVARIABLE	Environment variable not set
7184	LIC_ERR_FILENOTFOUND	License file not found
7185	LIC_ERR_FILESYNTAX	Error in license file
7186	LIC_ERR_TEXTIGNORE	Ignoring text in license file
7187	LIC_ERR_VMDetect	Component not licensed under virtualization
7188	LIC_ERR_VMCORES	The number of licensed cores is less than the actual cores

Error codes 7180 to 7186 are licensing errors. If you encounter any of these errors, rerun `vlicvrfy` utility to get further details about the cause of the error. If required, contact Versant for new license key for the component.

7200	UT_W_NOT_LATEST	There is a more recent backup of this level
7203	UT_W_NEWER_LEVEL_2	Level 2 more recent than level 1, restore level 2 again afterward
7204	UT_W_DB_MODIFIED	Db modified since last restore
7205	UT_W_END_OF_BACKUP	End of backup
7206	UT_DB_NOT_IN_DBID	Database not in osc-dbid file
7207	UT_ABNORMAL_EXIT	Utility process was killed or crashed. Command was \$(command)'.
7208	UT_ER_VBACKUP_FILE_READ	Cannot read vbackup status file for specified database
7209	UT_ER_VBACKUP_FILE_WRITE	Cannot write to vbackup status file for specified database
7210	UT_ER_VBACKUP_FILE_NO_ACCESS	Cannot access vbackup status file for specified database
7220	UT_NO_REPLICA_ENTRY	No entry for database in VERSANT_ROOT/replica.
7221	UT_IS_ALREADY_A_REPLICA_DB	Database is already replicated
7222	UT_MULTIPLE_REPLICA_INFOS	Multiple instances of replica_info were detected
7223	UT_FAILED_TO_CLONE_OBJECTS	Could not replicate all the objects
7224	UT_OBJECT_NOT_IN_DB1	Object '%s' is in '%s' but not '%s'
7225	UT_OBJECT_NOT_IN_DB2	Object '%s' is in '%s' but not '%s'
7226	UT_OBJECT_VALUES_DIFFERENT	Object '%s' has different values in '%s' and %s'
7227	UT_LOGGING_REQUIRED	CREATEREPLICA requires that logging be ON.
7228	UT_GROUP_REQUIRED	Replication does not support personal databases
7229	UT_USE_REMOVE_REPLICA	Use REMOVE_REPLICA to remove a replicated database

---

7230	UT_ER_DBID_IN_USE	Attempt to assign a dbid that is already in use.
7231	UT_ER_NOT_SUPER_USER	Must be super user for the operation
7232	UT_OPEN_REPLICA_FILE_FAIL	Attempt to open replica file failed
7233	UT_ER_ALREADYON	Roll forward already on
7234	UT_ER_ALREADYOFF	Roll forward already off
7235	UT_ER_INVALIDARG	Invalid argument to RF
7236	UT_ER_ARCHIVING_OFF	Archiving not turned on
7237	UT_ER_BACKUP_OFF	Backup not started
7238	UT_ER_BACKUP_NOCONFIRM	Prev backup not confirmed
7239	UT_ER_WRONGID	Confirmation for wrong ID
7240	UT_ER_ARCHIVING_PENDING	Archiving not turned on
7241	UT_ER_RECOVERY_OFF	RF recovery not started
7242	UT_ER_RFREC_TOO_LARGE	RF Record > 256K len
7243	UT_ER_RF_INVALIDREC	Rec-aligned RF rec needed
7244	UT_CANT_CONNECT_PDB	Cannot connect to personal database
7245	UT_DBA_PRIVILEGE	You need dba privilege to do this operation
7246	UT_ER_TERM_LOGGER	Logging process terminated
7247	UT_ER_ARCHIVE_REP	Cannot archive a replicated database
7248	UT_ER_INVALID_SEQNO	Out-of-order record seqno
7250	UT_FTS_INVALID_PDELAY	Invalid polling delay. Must be either 0 or -1
7251	UT_ER_BACKUP_NO_LEVEL_0	No level 0 backup
7252	UT_DB_NOT_EMPTY	Destination database must be empty
7253	UT_DB_WRONG_STATE	Database is in the wrong state
7254	UT_DB_ALREADY_SUSPENDED	Ftstool -stopsync has already been applied on the database
7255	MSG_VERSANT_UTILITY	VERSANT Utility \$(utility) Version \$(version) \$(comment)

---

7256	MSG_VERSANT_COPYRIGHT	Copyright (c) \$(from_year)-\$(current_year) VERSANT Corporation
7257	MSG_PANIC	Versant panicking in file '%s' at line %d
7258	MSG_YES	Yes
7259	MSG_NO	No
7260	MSG_ERROR	Error
7261	MSG_WARNING	Warning
7262	MSG_ABORT_ACTION	Do you want to abort?
7263	MSG_ANSWER_YES	Y
7264	MSG_ANSWER_NO	N
7275	MSG_UDLTOOL_USAGE	%% %s [-d ddbname] [-l sdbname] [-n] [-p] [-s [lm]] [filename]
7276	MSG_UDLTOOL_USAGE_1	-d ddbname db name where the lock model will be set
7277	MSG_UDLTOOL_USAGE_2	-l sdbname db name where the lock model object is/will be located
7278	MSG_UDLTOOL_USAGE_3	-n store the new lock model object into sdbname
7279	MSG_UDLTOOL_USAGE_4	-p print info of the lock model currently used in ddbname
7280	MSG_UDLTOOL_USAGE_5	-s [lm] set the lock model lm in sdbname to be used in ddbname
7281	MSG_UDLTOOL_USAGE_6	Filename name of the input file that describes the lock model
7282	MSG_UDLTOOL_USAGE_7	If -n is specified, both filename and sdbname are required
7283	MSG_UDLTOOL_USAGE_8	If -s or -p is specified, ddbname is required
7284	MSG_UDLTOOL_USAGE_9	If -s is specified and if sdbname is not provided, ddbname will be used
7285	MSG_UDLTOOL_USAGE_10	To retrieve the lock model object

---

---

7286	MSG_UDLTOOL_USAGE_11	If -s is specified, the lock model name can be obtained either from the
7287	MSG_UDLTOOL_USAGE_12	Command line or from the input file
7288	MSG_UDLTOOL_USAGE_13	If -s is specified and the lock model name is not provided, the Versant
7289	MSG_UDLTOOL_USAGE_14	Standard Lock Model will be used
7290	MSG_UDLTOOL_USAGE_15	Ddbname can be specified in the environment variable O_DBNAME
7291	MSG_UDLTOOL_USAGE_END	End of UDL help
7292	MSG_UDL_PARSE_ERROR	UDL Parser [line %d column %d]
7293	MSG_UDL_COMMENT_ERROR	UDL Parser: at line %d: no comment delimiter
7295	MSG_ADDVOL_USAGE	Usage: addvol parameters [options] <dbname>
7296	MSG_ADDVOL_USAGE_1	Parameters
7297	MSG_ADDVOL_USAGE_2	-n volname logical name for volume
7298	MSG_ADDVOL_USAGE_3	-p volpath path of volume device/file
7299	MSG_ADDVOL_USAGE_4	Options
7300	MSG_ADDVOL_USAGE_5	-s volsize volume size, default to 128M
7301	MSG_ADDVOL_USAGE_6	-e extsize extent size
7302	MSG_ADDVOL_USAGE_7	-i pre-allocate and initialize volume
7303	MSG_ADDVOL_USAGE_8	-noprint suppress display messages
7304	MSG_ADDVOL_USAGE_9	Ddbname: database name
7305	MSG_ADDVOL_USAGE_END	Usage: Addvol
7315	MSG_CREATEDB_USAGE	Usage: createdb [options] <dbname>
7316	MSG_CREATEDB_USAGE_1	Options:
7317	MSG_CREATEDB_USAGE_2	-i pre-allocate and initialize the database

---

---

7318	MSG_CREATEDB_USAGE_3	-il pre-initialize the physical and logical logs to logvol-maxsize
7319	MSG_CREATEDB_USAGE_4	-noprint suppress display messages
7320	MSG_CREATEDB_USAGE_5	Dbname: database name
7321	MSG_CREATEDB_USAGE_END	Usage: createdb
7325	MSG_DBUSER_USAGE	Usage: dbuser <command> [options] <dbname>
7326	MSG_DBUSER_USAGE_1	Command:
7327	MSG_DBUSER_USAGE_2	-list list database user access list (default)
7328	MSG_DBUSER_USAGE_3	-add add user to database access list
7329	MSG_DBUSER_USAGE_4	'-n' or '-P' options required , '-passwd' option maybe specified
7330	MSG_DBUSER_USAGE_5	-delete delete user from access list
7331	MSG_DBUSER_USAGE_6	'-n' or '-P' options required
7332	MSG_DBUSER_USAGE_7	-chpasswd change the password for an existing user
7333	MSG_DBUSER_USAGE_8	'-n' options required '-opasswd' and '-passwd' options may be specified
7334	MSG_DBUSER_USAGE_9	Options:
7335	MSG_DBUSER_USAGE_10	-n name user name
7336	MSG_DBUSER_USAGE_11	-passwd password password
7337	MSG_DBUSER_USAGE_12	-opasswd old-password old password
7338	MSG_DBUSER_USAGE_13	-P public access
7339	MSG_DBUSER_USAGE_14	-m r   rw access mode
7340	MSG_DBUSER_USAGE_15	-noprint suppress display messages
7341	MSG_DBUSER_USAGE_16	Dbname: database name
7342	MSG_DBUSER_USAGE_END	Usage: dbuser
7345	MSG_MAKEDB_USAGE	Usage: makedb [options] <dbname>

---



---

7346	MSG_MAKEDB_USAGE_1	Options:
7347	MSG_MAKEDB_USAGE_2	[ -g   -p ]
7348	MSG_MAKEDB_USAGE_3	-g    group database
7349	MSG_MAKEDB_USAGE_4	-p    personal database
7350	MSG_MAKEDB_USAGE_5	-owner user    make user as owner of database directory
7351	MSG_MAKEDB_USAGE_6	-cpprofile db    copy profiles from the specified db
7352	MSG_MAKEDB_USAGE_7	Directory
7353	MSG_MAKEDB_USAGE_8	-nofeprofile    do not create front-end profile
7354	MSG_MAKEDB_USAGE_9	-logging        turn logging on
7355	MSG_MAKEDB_USAGE_10	-locking        turn locking on
7356	MSG_MAKEDB_USAGE_11	-sglatch        turn multi_latch off
7357	MSG_MAKEDB_USAGE_12	-noprint        suppress display mes- sages
7358	MSG_MAKEDB_USAGE_13	-promptpasswd    prompt for DBA's password
7359	MSG_MAKEDB_USAGE_14	-beprofile filepath -- copy server profile file from speci- fied location
7360	MSG_MAKEDB_USAGE_15	-feprofile filepath -- copy application profile file from specified location
7361	MSG_MAKEDB_USAGE_16	-feprofile and -nofeprofile -- cannot be specified together
7362	MSG_MAKEDB_USAGE_17	-feprofile and -beprofile -- cannot be used with -cpprofile
7363	MSG_MAKEDB_USAGE_18	Dbname: database name
7364	MSG_MAKEDB_USAGE_END	Usage: makedb
7365	MSG_DBLIST_USAGE	Usage: dblist [options]
7366	MSG_DBLIST_USAGE_1	Options:
7367	MSG_DBLIST_USAGE_2	-all    list all db in the system (default)
7368	MSG_DBLIST_USAGE_3	-owner owner    list db owned by owner

---

7369	MSG_DBLIST_USAGE_4	-d dbname list only the named database
7370	MSG_DBLIST_USAGE_5	-dir list all databases and empty
7371	MSG_DBLIST_USAGE_6	Database directories
7372	MSG_DBLIST_USAGE_7	-noprint suppress display messages
7373	MSG_DBLIST_USAGE_END	Usage: dblist
7375	MSG_DBINFO_USAGE	Usage: dbinfo <options> <dbname>
7376	MSG_DBINFO_USAGE_1	Options :
7377	MSG_DBINFO_USAGE_2	[ -m   -0   -1   -d ]
7378	MSG_DBINFO_USAGE_3	-m set multi-user mode
7379	MSG_DBINFO_USAGE_4	-0 set unstartable mode
7380	MSG_DBINFO_USAGE_5	-1 set DBA-only single-connection
7381	MSG_DBINFO_USAGE_6	-d set DBA-only multi-connection
7382	MSG_DBINFO_USAGE_7	-p print existing database mode
7383	MSG_DBINFO_USAGE_8	-c create a new .lock file
7384	MSG_DBINFO_USAGE_9	Dbname: database name
7385	MSG_DBINFO_USAGE_END	Usage: dbinfo
7387	MSG_REMOVEDB_USAGE	Usage: removedb [options] <dbname>
7388	MSG_REMOVEDB_USAGE_1	Options :
7389	MSG_REMOVEDB_USAGE_2	-f force removing database
7390	MSG_REMOVEDB_USAGE_3	-rmdir remove database directory
7391	MSG_REMOVEDB_USAGE_4	-noprint suppress display messages
7392	MSG_REMOVEDB_USAGE_5	Dbname: database name
7393	MSG_REMOVEDB_USAGE_END	Usage: removedb
7395	MSG_STARTDB_USAGE	Usage: startdb <dbname>
7396	MSG_STARTDB_USAGE_END	Usage: startdb
7403	MSG_STOPDB_USAGE	Usage: stopdb [options] <dbname>
7404	MSG_STOPDB_USAGE_1	Options :

---

7405	MSG_STOPDB_USAGE_2	[ -f   -s   -st ]
7406	MSG_STOPDB_USAGE_3	-f force killing all active transactions
7407	MSG_STOPDB_USAGE_4	-s wait for update rpcs to be done
7408	MSG_STOPDB_USAGE_5	-st wait for all active transactions to be done
7409	MSG_STOPDB_USAGE_6	-noprint suppress display messages
7410	MSG_STOPDB_USAGE_7	Dbname: database name
7411	MSG_STOPDB_USAGE_END	Usage: stopdb
7415	MSG_MAKEPROFILE_USAGE	Usage: makeprofile [options] <dbname>
7416	MSG_MAKEPROFILE_USAGE_1	Options :
7417	MSG_MAKEPROFILE_USAGE_2	-cpprofile db copy profiles from the specified db
7418	MSG_MAKEPROFILE_USAGE_3	Directory
7419	MSG_MAKEPROFILE_USAGE_4	-nofeprofile do not create front-end profile
7420	MSG_MAKEPROFILE_USAGE_5	-logging turn logging on
7421	MSG_MAKEPROFILE_USAGE_6	-locking turn locking on
7422	MSG_MAKEPROFILE_USAGE_7	-sglatch turn multi_latch off
7423	MSG_MAKEPROFILE_USAGE_8	-noprint suppress display messages
7424	MSG_MAKEPROFILE_USAGE_9	Dbname: database name
7425	MSG_MAKEPROFILE_USAGE_END	Usage: makeprofile
7426	MSG_DBID_USAGE	Usage: dbid [options] <dbname>
7427	MSG_DBID_USAGE_1	Options :
7428	MSG_DBID_USAGE_2	-N Create an osc-dbid file. <dbname> is not specified
7429	MSG_DBID_USAGE_3	-c Create an entry in osc-dbid file for database <dbname>
7430	MSG_DBID_USAGE_4	-C <dbid> Create an entry in osc-dbid file for database <dbname>
7431	MSG_DBID_USAGE_5	And database id as <dbid>.

---

7432	MSG_DBID_USAGE_6	-t <type> Specify the type of database (group or personal )
7433	MSG_DBID_USAGE_7	Must be used with -c or -C options.
7434	MSG_DBID_USAGE_8	Default database type is group.
7435	MSG_DBID_USAGE_9	-d Remove the database <dbname> entry from the osc-dbid file
7436	MSG_DBID_USAGE_10	-r Print the database id of the database <dbname>
7437	MSG_DBID_USAGE_11	-noprint Suppress display messages
7438	MSG_DBID_USAGE_12	Dbname: database name
7439	MSG_DBID_USAGE_13	Type: database type
7440	MSG_DBID_USAGE_14	1: Personal database
7441	MSG_DBID_USAGE_15	2: Group database
7442	MSG_DBID_USAGE_END	Usage: dbid
7445	MSG_REORGDB_USAGE	Usage: reorgdb parameters <dbname>
7446	MSG_REORGDB_USAGE_1	Parameters :
7447	MSG_REORGDB_USAGE_2	Options :
7448	MSG_REORGDB_USAGE_3	-noprint suppress display messages
7449	MSG_REORGDB_USAGE_4	Dbname: database name
7450	MSG_REORGDB_USAGE_END	Usage: reorgdb
7455	MSG_RESTOREDB_USAGE	Usage: restoredb parameters <dbname>
7456	MSG_RESTOREDB_USAGE_1	Parameters :
7457	MSG_RESTOREDB_USAGE_2	-if device input device name
7458	MSG_RESTOREDB_USAGE_3	Options :
7459	MSG_RESTOREDB_USAGE_4	-nowait don't wait for operator assistance
7460	MSG_RESTOREDB_USAGE_5	-noprint suppress display messages
7461	MSG_RESTOREDB_USAGE_6	Dbname: database name
7462	MSG_RESTOREDB_USAGE_END	Usage: restoredb

---

---

7465	MSG_OSCP_USAGE	Usage: oscp [options]
7466	MSG_OSCP_USAGE_1	Options :
7467	MSG_OSCP_USAGE_2	-d database path
7468	MSG_OSCP_USAGE_3	-i environment information
7469	MSG_OSCP_USAGE_4	-l product version number with patch
7470	MSG_OSCP_USAGE_5	-n dbid node name
7471	MSG_OSCP_USAGE_6	-o osc-dbid path
7472	MSG_OSCP_USAGE_7	-p Versant runtime path
7473	MSG_OSCP_USAGE_8	-r Versant root path
7474	MSG_OSCP_USAGE_9	-v product version number
7475	MSG_OSCP_USAGE_10	-b patch information
7476	MSG_OSCP_USAGE_END	Usage: oscp
7477	MSG_SSD_USAGE	Usage: ss.d <command> [options]
7478	MSG_SSD_USAGE_1	Command:
7479	MSG_SSD_USAGE_2	-start start the system services daemon, requires '-port' option
7480	MSG_SSD_USAGE_3	Options:
7481	MSG_SSD_USAGE_4	-port port-number port number where ss.d has to accept new versant connection
7482	MSG_SSD_USAGE_5	-log log-file-name logs additional information to specified log file
7483	MSG_SSD_USAGE_END	Usage: ss.d
7495	MSG_CONVERTDB_USAGE	Usage: convertddb <dbname>
7496	MSG_CONVERTDB_USAGE_1	Options:
7497	MSG_CONVERTDB_USAGE_2	-noprint suppress display messages
7498	MSG_CONVERTDB_USAGE_END	Usage: convertddb
7505	MSG_DBUSER_NAME	User Name
7506	MSG_DBUSER_READ_ACCESS	R
7507	MSG_DBUSER_WRITE_ACCESS	W
7508	MSG_DBUSER_DBA	DBA

---

7509	MSG_DBUSER_CREATION_DATE	Creation Date
7515	MSG_DBID_HEADER_INFO	DBID file header
7516	MSG_DBID_MAX_ID	Max ID = %d
7517	MSG_DBID_VERSION	Version = %d
7518	MSG_DBID_VERSION2	ersion2 = %d
7519	MSG_DBID_ENTRIES	Entries = %d
7520	MSG_DBID_VALID_ENTRIES	Valid entries = %d
7521	MSG_DBID_PATH	Path = %s
7525	MSG_DBLIST_ENTRY	ID = %d
7526	MSG_DBLIST_ENTRY_EMPTY	ID = EMPTY DB DIREC- TORY
7527	MSG_DBLIST_DBNAME	DB name = %s
7528	MSG_DBLIST_CREATOR	Creator = %s
7529	MSG_DBLIST_DATE_CREATED	Date created = %s
7530	MSG_DBLIST_TYPE_PERSONAL	Db type = PERSONAL DATABASE
7531	MSG_DBLIST_TYPE_GROUP	Db type = GROUP DATABASE
7532	MSG_DBLIST_TYPE_UNIVERSAL	Db type = PUBLIC DATABASE
7533	MSG_DBLIST_TYPE_UNKNOWN	Db type = UNKNOWN
7534	MSG_DBLIST_VERSION	Db version = %s
7535	MSG_DBLIST_GROUPID	Db group id = %d
7545	MSG_DBID_ENTRY	DBID entry %d
7546	MSG_DBID_ENTRY_EMPTY	ID = EMPTY DB DIRECTORY
7547	MSG_DBID_ID	ID = %d
7548	MSG_DBID_DBNAME	DB name = %s
7549	MSG_DBID_CREATOR	Creator = %s
7550	MSG_DBID_DATE_CREATED	Date created = %s
7551	MSG_DBID_DBVERSION	Version = %s
7555	MSG_VERSION	Versant Product Version: %s
7556	MSG_ROOT_PATH	Versant Root Path: %s
7557	MSG_ROOT_NOACCESS	WARNING: no access to root path
7558	MSG_TOO_MANY_LINK_LEVELS	WARNING: too many level of links

---

7564	MSG_DB_ENV	VERSANT_DB: %s
7565	MSG_DBID_ENV	VERSANT_DBID: %s
7566	MSG_REL_ENV	VERSANT_REL: %s
7567	MSG_SSD_NOACCESS	WARNING: CANNOT LOCATE ss.d
7568	MSG_SSD	Versant ss.d Location: %s
7569	MSG_DBID_NODE	Versant osc-dbid node name: %s
7570	MSG_DBID_PATH_IS	Versant osc-dbid path: %s
7575	MSG_ADD_USER	Adding User Name = %s
7576	MSG_ADD_USER_INFO	User Inserted
7577	MSG_DELETE_USER	Deleting User Name = %s
7578	MSG_DELETE_USER_INFO	User Deleted
7585	MSG_WRONG_VOLUME_MOUNTED	WRONG VOLUME SEQUENCE NUMBER
7586	MSG_CHECK_VOLUME_NUMBER	ATTENTION: Is new tape # %d mounted and ready?
7587	MSG_VOLUME_INFO	Volume %s at %s, size=%d
7595	MSG_REORGDB_INFO	Reorganizing database
7596	MSG_REORGDB_DEVICE	Device path = %s
7605	MSG_RESTOREDB_INFO	Restoring database
7606	MSG_RESTOREDB_NAME	Database name = %s
7607	MSG_RESTOREDB_VOLUMES	Total volumes = %d
7615	MSG_COMPRESSDB_INFO	Compress Database
7616	MSG_COMPRESSDB_FAILED	ERROR: Compress Database Failed
7617	MSG_COMPRESSDB_RESTORE	Restoring database
7625	MSG_REMOVEDB_WARNING	Attempt to force removal of database
7626	MSG_REMOVEDB_SYSVOL	Warning: system volume may be corrupted
7627	MSG_REMOVEDB_DATAVOL	User volumes created by addvol utility are not removed
7628	MSG_REMOVEDB_RETRY	Retrying (this may take up to 2 minutes)
7629	MSG_REMOVEDB_DBID	Database entry %d removed from ID file
7630	MSG_REMOVEDB_VOLNAME	Deleting volume at %s

---

7631	MSG_REMOVEDB_DBDIR	Removing database directory %s
7635	MSG_ADDVOL_INFO	Add Volume
7636	MSG_ADDVOL_NAME	Volume name = %s
7637	MSG_ADDVOL_SIZE	Volume size = %u pages (page size = %d bytes)
7638	MSG_ADDVOL_PATH	Volume path = %s
7639	MSG_ADDVOL_OFFSET	Volume offset = %d
7640	MSG_ADDVOL_EXTSIZE	Extent size = %d
7641	MSG_ADDVOL_NUMOFEXT	Number extents = %d
7642	MSG_ADDVOL_INSERT_ENTRY	Inserting volume entry
7645	MSG_CREATEDB_DBID	Database ID is %d
7646	MSG_CREATEDB_PLOG	Physical log volume created at %s
7647	MSG_CREATEDB_PLOG_INFO	Size=%u pages ( page size = %d bytes )
7648	MSG_CREATEDB_LLOG	Logical log volume created at %s
7649	MSG_CREATEDB_LLOG_INFO	Size=%u pages ( page size = %d bytes )
7650	MSG_CREATEDB_SYSVOL	System volume created at %s
7651	MSG_CREATEDB_SYSVOL_INFO	Size=%u pages ( page size=%d bytes ), extent size=%d, number of extents=%d
7652	MSG_CREATEDB_FORMAT	Formatting system volume
7653	MSG_CREATEDB_SYSCCLASS	Creating system classes
7655	MSG_STOPDB_WARNING	Database may be inconsistent while attempting to remove shared memory
7656	MSG_STOPDB_RETRY	Retrying (this may take up to 3 minutes)
7657	MSG_MAKE_DIRECTORY	Making directory at %s
7665	MSG_BEPROFILE	Back end profile created at %s
7666	MSG_FEPROFILE	Front end profile created at %s
7675	MSG_LOCK_MODES	%s Lock Model has %d lock modes
7676	MSG_LOCK_RS_TABLE	Relative Strength Table

---



---

7677	MSG_LOCK_COMP_TABLE	Compatibility Table
7678	MSG_LOCK_IPEL_TABLE	Implicit Promotion of Explicit Locks Table
7679	MSG_LOCK_PROMO_TABLE	Other Promotions Table
7685	MSG_MSG_NOTFOUND	Non-existent error message
7686	MSG_MSG_FILE_NOTFOUND	Can't find error message file error.msg
7687	MSG_MSG_CREATE_INDEX	Can't create error message index file error.msi
7688	MSG_MSG_WRITE_INDEX	Error while writing error message index file error.msi
7689	MSG_MSG_REMOVE_INDEX	Error while removing faulty error message index file error.msi
7690	MSG_MSG_SEEK_ERROR	Error while seeking on error message index file error.msi
7691	MSG_MSG_LINE_ERROR	Error: A line in error message file error.msg does not begin with a number
7695	MSG_DBMODE	Database is in %s mode
7696	MSG_CREATE_DBLOCK	Database lock file is created
7705	MSG_STARTDB_BEGIN_PHY_REC	Beginning physical recovery phase
7706	MSG_STARTDB_END_PHY_REC	Ending physical recovery phase
7707	MSG_STARTDB_BEGIN_LOG_REC	Beginning logical recovery phase
7708	MSG_STARTDB_END_LOG_REC	Ending logical recovery phase
7709	MSG_STARTDB_BEGIN_PREINIT_PLOG	Beginning physical log pre-initialization phase
7710	MSG_STARTDB_BEGIN_PREINIT_LLOG	Beginning logical log pre-initialization phase
7711	MSG_STARTDB_END_PREINIT_PLOG	Ending physical log pre-initialization phase
7712	MSG_STARTDB_END_PREINIT_LLOG	Ending logical log pre-initialization phase

---

7715	MSG_CREATEREPLICA_USAGE	Usage: createreplica [options] <dbname> <replica-dbname>
7716	MSG_CREATEREPLICA_USAGE_1	Options:
7717	MSG_CREATEREPLICA_USAGE_2	-i pre-allocate and initialize the database
7718	MSG_CREATEREPLICA_USAGE_3	-noprint suppress display mes- sages
7719	MSG_CREATEREPLICA_USAGE_4	-nocreate copy to existing data- base
7720	MSG_CREATEREPLICA_USAGE_5	-threads N fork N threads for copying
7721	MSG_CREATEREPLICA_USAGE_6	-batchsize M copy M objects per transaction default = 1000)
7722	MSG_CREATEREPLICA_USAGE_7	Dbname: name of database to rep- licate
7723	MSG_CREATEREPLICA_USAGE_8	Replica-dbname: database name for replica
7724	MSG_CREATEREPLICA_USAGE_9	-nocreate and -i cannot be used together
7725	MSG_CREATEREPLICA_USAGE_END	Usage: createreplica
7726	MSG_REMOVEREPLICA_USAGE	Usage: removereplica [options] <replica-dbname>
7727	MSG_REMOVEREPLICA_USAGE_1	Options:
7728	MSG_REMOVEREPLICA_USAGE_2	-f force removing of data- base
7729	MSG_REMOVEREPLICA_USAGE_3	-rmdir remove database direc- tory
7730	MSG_REMOVEREPLICA_USAGE_4	-noprint suppress display mes- sages
7731	MSG_REMOVEREPLICA_USAGE_5	Replica-dbname: name of replica database
7732	MSG_REMOVEREPLICA_USAGE_END	Usage: removereplica
7735	MSG_COMPAREDB_USAGE	Usage: comparedb [options] <db1> <db2>
7736	MSG_COMPAREDB_USAGE_1	Options:
7737	MSG_COMPAREDB_USAGE_2	-value compare object values

---

7738	MSG_COMPAREDB_USAGE_3	-noprint suppress display messages
7739	MSG_COMPAREDB_USAGE_4	-advanced -classes <class1> <class2> ... Compare only the listed classes
7740	MSG_COMPAREDB_USAGE_5	-classes <class1> <class2> ... Compare only the listed classes
7741	MSG_COMPAREDB_USAGE_6	-fullcompare list all the differences (option incompatible with -noprint)
7742	MSG_COMPAREDB_USAGE_7	Db1: name of first database
7743	MSG_COMPAREDB_USAGE_8	Db2: name of second database
7744	MSG_COMPAREDB_USAGE_END	Usage: comparedb
7745	MSG_SETDBID_USAGE	Usage: setdbid [options] <dbid> <dbname>
7746	MSG_SETDBID_USAGE_1	Options:
7747	MSG_SETDBID_USAGE_2	-noprint suppress display messages
7748	MSG_SETDBID_USAGE_3	Dbid: new dbid number
7749	MSG_SETDBID_USAGE_4	Dbname: name of the database
7750	MSG_SETDBID_USAGE_END	Usage: setdbid
7755	MSG_SETMINDBID_USAGE	Usage: setmindbid [options] <dbid>
7756	MSG_SETMINDBID_USAGE_1	Options:
7757	MSG_SETMINDBID_USAGE_2	-noprint suppress display messages
7758	MSG_SETMINDBID_USAGE_3	Dbid: new minimum dbid number
7759	MSG_SETMINDBID_USAGE_END	sage: setdbid
7765	MSG_CREATEREPLICA_DONE	The replica has been successfully created.
7766	MSG_REMOVE_REPLICA_DONE	The replica has been successfully removed.
7767	MSG_REPLICA_FILE_REMINDER	**REMEMBER TO UPDATE THE REPLICICA FILE**
7768	MSG_COMPAREDB_VALUE_SUCCESS	Success! Object loids and values are the same in both databases.

---

---

7769	MSG_COMPAREDB_LOID_SUCCESS	Success! Object loids are the same in both databases.
7770	MSG_COMPAREDB_DIFFERENCES	Some of the objects are different in two databases.
7775	MSG_ADDUSER_USAGE	Usage: adduser [options] <dbname>
7776	MSG_ADDUSER_USAGE_1	Options (either -n or -P is required)
7777	MSG_ADDUSER_USAGE_2	-n name            user name
7778	MSG_ADDUSER_USAGE_3	-P    public access
7779	MSG_ADDUSER_USAGE_4	-noprint        suppress display messages
7780	MSG_ADDUSER_USAGE_5	Dbname: database name
7781	MSG_ADDUSER_USAGE_END	Usage: adduser
7785	MSG_DELUSER_USAGE	Usage: deluser [options] <dbname>
7786	MSG_DELUSER_USAGE_1	Options (either -n or -P is required):
7787	MSG_DELUSER_USAGE_2	-n name            user name
7788	MSG_DELUSER_USAGE_3	-P                public access
7789	MSG_DELUSER_USAGE_4	-noprint        suppress display messages
7790	MSG_DELUSER_USAGE_5	Dbname: database name
7791	MSG_DELUSER_USAGE_END	Usage: deluser
7792	MSG_POLLING_BEGIN	Polling process %d begins
7793	MSG_POLLING_BEGINSESSION	Polling begin session on %s
7794	MSG_POLLING_ENDSESSION	Polling end session
7795	MSG_POLLING_CONNECTDB	Polling connectdb to %s
7796	MSG_POLLING_DISCONNECTDB	Polling disconnect from %s
7797	MSG_POLLING_SETDBUP	Polling set %s to UP
7798	MSG_POLLING_SETDBSYNC	Polling set %s to SYNC
7799	MSG_POLLING_RECURSIVE_FAIL	Polling gets recursive failure on %s
7800	MSG_POLLING_FAIL	Polling process fails on %s with err %d

---

7801	MSG_POLLING_EXIT_OK	Polling exits successfully
7802	MSG_POLLING_EXIT_FAIL	Polling exits with err %d
7803	MSG_POLLING_SETDB_FAIL	Polling set db status on %s fails with err %d
7804	MSG_POLLING_BEGINSESSION_FAIL	Polling begin session on %s fails with err %d
7805	MSG_POLLING_ENDSESSION_FAIL	Polling end session fails with err %d
7806	MSG_POLLING_CONNECTDB_FAIL	Polling connectdb fails with err %d
7807	MSG_POLLING_BEGIN_SYNC	Polling begins to do the sync from %s to %s
7808	MSG_POLLING_SYNC_FAIL	Polling sync fails with err %d
7809	MSG_POLLING_DISCONNECTDB_FAIL	Polling disconnectdb fails with err %d
7810	MSG_POLLING_SET_UNSTARTABLE	Polling has set %s to unstartable mode automatically
7811	MSG_POLLING_SET_MULTIUSER	Polling has set %s back to multi-user mode
7812	MSG_POLLING_USAGE	Usage: polling <fromdb> <todb>
7813	MSG_POLLING_COMMIT	Polling does a commit
7814	MSG_POLLING_COMMIT_FAIL	Polling commit fails with err %d
7815	MSG_POLLING_SYNC_OK	Polling finishes the sync successfully
7816	MSG_POLLING_RUNNING	Polling already running
7817	MSG_POLLING_VANISH	Polling disappeared
7818	MSG_VCOPYDB_USAGE	Usage: vcopydb [options] <dbname> <replica-dbname>
7819	MSG_VCOPYDB_USAGE_1	Options:
7820	MSG_VCOPYDB_USAGE_2	-nolock ignore locks in source db
7821	MSG_VCOPYDB_USAGE_3	-i pre-allocate and initialize the database
7822	MSG_VCOPYDB_USAGE_4	-noprint suppress display messages

---

7823	MSG_VCOPYDB_USAGE_5	-nocreate      copy to existing database
7824	MSG_VCOPYDB_USAGE_6	-threads N      fork N threads for copying
7825	MSG_VCOPYDB_USAGE_7	-batchsize M    copy M objects per transaction (default = 1000)
7826	MSG_VCOPYDB_USAGE_8	Dbname: name of database to replicate
7827	MSG_VCOPYDB_USAGE_9	Replica-dbname: database name for replica
7828	MSG_VCOPYDB_USAGE_10	nocreate and -i cannot be used together
7829	MSG_VCOPYDB_USAGE_END	Usage: vcopydb
7832	MSG_POLLING_SETTING_MULTUSER	Polling is setting %s back to multi-user mode
7835	MSG_FTSTOOL_USAGE	Usage: ftstool [options] <dbname>
7836	MSG_FTSTOOL_USAGE_1	Options:
7837	MSG_FTSTOOL_USAGE_2	-p T      fork a polling process after T seconds
7838	MSG_FTSTOOL_USAGE_3	T can be either:
7839	MSG_FTSTOOL_USAGE_4	: no delay, or
7840	MSG_FTSTOOL_USAGE_5	-1: disable automatic forking
7841	MSG_FTSTOOL_USAGE_6	-disable/-forcedisable
7842	MSG_FTSTOOL_USAGE_7	-enable
7843	MSG_FTSTOOL_USAGE_8	-stopsync [-batchsize N] change database to SUSPEND state
7844	MSG_FTSTOOL_USAGE_9	Polling records are discarded in batches: default N = 1000
7845	MSG_FTSTOOL_USAGE_10	-status
7846	MSG_FTSTOOL_USAGE_11	Dbname: database name
7847	MSG_FTSTOOL_USAGE_END	Usage: ftstool
7848	MSG_FTSTOOL_CURRENTPD	Current polling delay: %d
7849	MSG_FTSTOOL_SETPD	Setting polling delay to %d

---

7850	MSG_FTSTOOL_SUSPEND	Database state changed to SUSPEND
7855	MSG_VERSANT_RELEASE	Versant Release Version: %s
7856	MSG_PLATFORM	Platform: %s
7857	MSG_OS	Operating System: %s
7858	MSG_OS_PATCH	OS Patch Info: %s
7859	MSG_C	C Version: %s
7860	MSG_CPLUSPLUS	C++ Version: %s
7861	MSG_CPLUSPLUS_PATCH	C++ Patch Info: %s
7865	MSG_VARCOMPARE_USAGE	Usage: varcompare -classes <class1> <class2> ... <db1> <db2>
7866	MSG_VARCOMPARE_USAGE_1	Db1: name of first database
7867	MSG_VARCOMPARE_USAGE_2	Db2: name of second database
7868	MSG_VARCOMPARE_USAGE_END	Usage: varcompare
7869	MSG_VMOVEDB_USAGE	Usage: vmovedb [options] [-C c1 c2 c3 ...] <source> <target>
7870	MSG_VMOVEDB_USAGE_1	Options:
7871	MSG_VMOVEDB_USAGE_2	-subclasses select all sub- classes for the specified classes
7872	MSG_VMOVEDB_USAGE_3	-closure L move objects in clo- sure set for L levels
7873	MSG_VMOVEDB_USAGE_4	-threads N fork N threads in vmovedb
7874	MSG_VMOVEDB_USAGE_5	batchsize S move S objects per transaction (default = 1000)
7875	MSG_VMOVEDB_USAGE_6	-C <c1> <c2> <c3> .... Select objects from classes c1, c2, c3 only (default = all user defined classes)
7876	MSG_VMOVEDB_USAGE_7	Source: name of source database
7877	MSG_VMOVEDB_USAGE_8	Target: name of the target data- base
7878	MSG_VMOVEDB_USAGE_END	Usage: vmovedb
7879	MSG_VMOVEDB_DONE	Vmovedb completed successfully

---

---

7880	MSG_GETVBACKUPSTATUS_USAGE	Usage: getvbackupstatus <dbname>
7881	MSG_GETVBACKUPSTATUS_USAGE _END	Usage: getvbackupstatus
7882	MSG_GETVBACKUPSTATUS_DONE	Getvbackupstatus executed suc- cessfully
7885	MSG_VLICVRFY_USAGE	Usage: vlicvrfy [ <license-file> ]
7886	MSG_VLICVRFY_USAGE_END	Usage: vlicvrfy
7887	MSG_VLICCHK_USAGE	Usage: vlicchk -c <component> -v <version> [<license-server>]
7888	MSG_VLICCHK_USAGE_END	Usage: vlicchk
9601	EVJ_BAD_SESSION	Bad session, or session not begun, or session ended
9602	EVJ_JAVA_CTOR	Constructor failed for a java class
9603	EVJ_BAD_LOID_STRING	String not valid for conversion to Handle
9604	EVJ_THREW_ERROR	Internal error
9605	EVJ_BOMB	Internal error
9606	EVJ_OUT_OF_MEMORY	Out of Memory
9607	EVJ_BAD_UNICODE_UTF	Cannot convert UTF characters to unicode
9608	EVJ_NO_SUCH_ATTR	Attribute does not exist
9609	EVJ_BAD_ATTR_TYPE	Attribute is wrong type
9610	EVJ_NOT_IMPLEMENTED	Feature not implemented
9611	EVJ_NOT_CLASS_OBJ	Operation requires a class object
9612	EVJ_BAD_CAPABILITY	Valid newsessioncapability required
9613	EVJ_ALREADY_CAPABILITY	Only one newsessioncapability can be created
9614	EVJ_NOT_HANDLE_OBJ	Handlevector contains a non-han- dle object
9615	EVJ_CANNOT_GETPUT_PATH_ATTR	Cannot get/put on an attribute object for path query only



---

9616	EVJ_CANNOT_GETPUT_EMPTY_HANDLE	Cannot get/put on an empty handle which does not correspond to any database objects
9617	EVJ_CANNOT_END_SESSION_INUSE	Cannot end a session which is in use
9618	EVJ_CANNOT_END_UNATTACHED_SESSION	Cannot end a session which is not attached by this thread
9619	EVJ_QUERY_PARAMETER_COUNT_INVALID	Any parameters specified in VQL query string are not set by a call to bind
9620	EVJ_HANDLE_OPERATION_INVALID	The operation on the handle is invalid because it contains a Smalltalk constant
9621	EVJ_SESSION_MISMATCH	The session of the current thread is different from the session of the persistent object
9622	EVJ_CURSOR_VQLQUERY_INVALID	This type of vqlquery String is not supported for cursor based queries
9800	EVJ_METHOD_NOT_SUPPORTED	The method is not implemented in this class
9801	EVJ_NOT_A_PERSISTENT_CLASS	The class of the object does not belong to a persistent category
9802	EVJ_PARAMETER_INVALID	The specified parameter is invalid
9803	EVJ_SCHEMA_JAVA_DB_TYPE_MISMATCH	The database schema differs from the Java class definition with respect to the type of an attribute
9804	EVJ_SCHEMA_JAVA_ATTR_MISSING	The database schema differs from the Java class definition because attribute present in the database schema is missing from the Java class definition
9805	EVJ_SCHEMA_DB_ATTR_MISSING	The database schema differs from the Java class definition because an attribute present in the Java class definition does not exist in the database schema

9806	EVJ_SERIALIZATION_EXCEPTION	Serialization of an object failed.
9807	EVJ_DESERIALIZATION_EXCEPTION	Deserialization of an object failed
9808	EVJ_ROOTNAME_ALREADY_DEFINED	A root object with the given name already exists in the database
9809	EVJ_ROOTNAME_DOES_NOT_EXIST	A root object with the given name does not exist in the database
9810	EVJ_NOT_A_VALID_KEY_TYPE	The type of given key for VQL query is not supported.
9811	EVJ_NOT_A_PERSISTENT_INSTANCE	The instance object has not been made persistent yet
9812	EVJ_OBJECT_WAS_CLEARED_FROM _JAVA_CACHE	The object was cleared from the Java cache and is no longer valid
9813	EVJ_EXCEPTIONS_ACCESSING _ARRAY	There were exceptions when accessing an array of persistent objects. The exceptions are stored in the vexception arguments
9814	EVJ_NOT_IN_SESSION	The current thread is not in a session
9815	EVJ_TOO_MANY_SESSIONS	The session pool has exceeded its limit on number of sessions
9816	EVJ_NOT_A_TRANSIENT_INSTANCE	The instance object has been made persistent
9817	EVJ_OBJECT_IS_ACTIVE	The persistent object is still active within its transaction
9818	EVJ_SESSION_CLASSLOADER_USED _WITH_LAUNCHER	Cannot call setsessionclassloader method while using the Launcher
9819	EVJ_OBJECT_DOES_NOT_HAVE_OID	Cannot call setpersistentstate on a transient object that does not have an oid associated with it
9820	EVJ_OBJECT_HAS_OID	Cannot call makepersistent on a transient object that has an oid associated with it

---

9821	EVJ_OBJECT_ALWAYS_PERSISTENT	Attempted to make an always persistent object transient.
9822	EVJ_INVALID_OID_SPECIFIED	The oid specified for the transient object already exists. Cannot call setpersistentstate with the option NEW_STATE for this transient object
9823	EVJ_INVALID_USERNAME	The user name is limited to 35 characters
9824	EVJ_INVALID_PASSWORD	The user password is limited to 256 characters
9825	EVJ_INTERNAL_ERROR	Number format Exception.
9826	EVJ_INVALID_CACHED_OBJECTS_STATE	The state specified for cached objects is invalid
9827	EVJ_INVALID_SESSION	The session of the current thread and the session in which the query was executed are different
9828	EVJ_CANNOT_GET_CLASS_OBJECT_FOR_EMPTY_HANDLE	Cannot get a class object for an empty handle which does not correspond to any database object
11001	VUTIL_BACKEND_PROFILE__CLASSNAME	Server Process Parameters -- The name of the backendprofile class
11002	VUTIL_BACKEND_PROFILE__READ_FAILED	Error reading Server Process parameter file -- Error when read fails
11003	VUTIL_BACKEND_PROFILE__WRITEFAILED	Error writing Server Process parameter file -- Error when write fails
11051	VUTIL_PROFILE__CLASSNAME	Parameters -- The name of the Profile class
11052	VUTIL_PROFILE__UNSPECIFIED	Default -- The word to use for unspecified parameters
11053	VUTIL_PROFILE__WAIT_FOREVER	Wait Forever -- The word to use for wait-forever parameters
11054	VUTIL_PROFILE__UNSPECIFY_OP	Set to 'Default' -- The name of the Profile_unspecify operation

---

11055	VUTIL_PROFILE__WAIT_FOREVER__OP	Set to 'Wait Forever' -- The name of the Profile_waitforever operation
11076	VUTIL_COMMANDS__INVALID_MODE	Invalid Mode -- Bad value for mode argument
11077	VUTIL_COMMANDS__POPEN_FAILED	Could not open pipe -- Popen failed
11078	VUTIL_COMMANDS__OS_ERROR	Operating System Error %s
11101	VUTIL_DATABASE__CLASSNAME	Database -- The name of the Database class
11102	VUTIL_DATABASE__START	Start -- The name of the Database_startdb operation
11103	VUTIL_DATABASE__STOP	Stop -- The name of the Database_stopdb operation
11104	VUTIL_DATABASE_STOP__OPTIONS	Force to stop : Arg 1 -- options for stopdb: force or don't force
11105	VUTIL_DATABASE_STOP__FORCE	Force : Arg 1, option 1 -- force
11106	VUTIL_DATABASE_STOP__DONT_FORCE	Don't Force : Arg 1, option 2 -- don't force
11107	VUTIL_DATABASE_STOP__MESSAGE	Force killing of all active transactions? -- The message displayed in the operation dialog box
11108	VUTIL_DATABASE__DESTROY	Remove -- The name of the Database_destroydatabases operation
11109	VUTIL_DATABASE_DESTROY__OPTIONS	Force to remove : Arg 1 -- options for removedb: force or don't force
11110	VUTIL_DATABASE_DESTROY__FORCE	Force : Arg 1 option 1 -- force
11111	VUTIL_DATABASE_DESTROY__DON'T_FORCE	Don't Force : Arg 1 option 2 -- don't force
11112	VUTIL_DATABASE_DESTROY__MESSAGE	Are you sure you want to destroy all objects in %s? -- The message displayed in the operation dialog box

---

11113	VUTIL_DATABASE__ADDVOL	Add Volume -- The name of the Database_addvolume operation
11114	VUTIL_DATABASE_ADDVOL__PATH	Path of device/file : Arg 1 -- the path of the device/file
11115	VUTIL_DATABASE_ADDVOL__SIZE	Volume size : Arg 2 -- the size of the new volume
11116	VUTIL_DATABASE_ADDVOL__EXTENT__SIZE	Pages per extent : Arg 3 -- the extent size of the new volume
11117	VUTIL_DATABASE__ADDUSER	Add The name of the Database_adduser operation
11118	VUTIL_DATABASE__REMOVEUSER	Remove Selection The name of the Database_removeuser operation
11119	VUTIL_DATABASE__SCH2DB	Load Schema File The name of the Database_sch2db operation
11120	VUTIL_DATABASE_SCH2DB__FILE	Schema file(s) : Arg 1 -- the schema file or files
11121	VUTIL_DATABASE_SCH2DB__ON__CONFLICT	On Conflict : Arg 2 -- how to handle a conflict
11122	VUTIL_DATABASE_SCH2DB__ABORT	Abort : Arg 2,option 1 -- abort the operation
11123	VUTIL_DATABASE_SCH2DB__DROP	Drop : Arg 2, option 1 -- drop the classes
11124	VUTIL_DATABASE_SCH2DB__EVOLVE	Evolve : Arg 2, option 1 -- evolve the classes
11125	VUTIL_DATABASE_SCH2DB__MESSAGE	Are sure you want to make these changes? --The message displayed in the dialog box
11126	VUTIL_DATABASE_SCH2DB__UNNECESSARY	No changes are necessary -- The message displayed when no changes are necessary
11127	VUTIL_DATABASE__DROPCLASS	Remove Selection -- The name of the Database_removeclasses operation
11128	VUTIL_DATABASE_DROPCLASS__MESSAGE	Drop the following classes? -- The message to show in the dialog box

---

11129	VUTIL_DATABASE__DROPINST	Drop Instances of Selection -- The name of the Database_removeinstances operation
11130	VUTIL_DATABASE_DROPINST __MESSAGE	Remove instance of the following classes? -- The message to show in the dialog box
11131	VUTIL_DATABASE__SYNC_CLASS	Copy Class -- The name of the Database_syncclass operation
11132	VUTIL_DATABASE_SYNC_CLASS __CLASS	Source Class : Arg 1 -- The class to sync to
11133	VUTIL_DATABASE__SYNC_CLASS _QUERY	Copy Class -- The name of the Database_syncclassquery operation
11134	VUTIL_DATABASE_SYNC_CLASSQUER Y__CLASS	Source Class : Arg 1 -- The class to sync to
11135	VUTIL_DATABASE_SYNC_CLASS _QUERY__CONFLICT	On conflict : Arg 2 -- how to deal with a conflict
11136	VUTIL_DATABASE_SYNC_CLASS _QUERY__DROP	Drop arg 2 -- option 1 -- drop instances
11137	VUTIL_DATABASE_SYNC_CLASS _QUERY__PRESERVE	Evolve arg 2 -- option 2 -- evolve instances
11138	VUTIL_DATABASE_SYNC_CLASS _QUERY__MESSAGE	A conflict exists -- The message in the dialog box
11139	VUTIL_DATABASE__SHOW_ALL _CLASSES	Show All Classes -- The name of the Database_showallclasses operation
11140	VUTIL_DATABASE__SHOW _SUBCLASSES	Show Immediate Subclasses Of : - - The name of the Database_showsubclassesof operation
11141	VUTIL_DATABASE_SHOW _SUBCLASSES__CLASS	Subclasses Of : arg 1 -- the class
11142	VUTIL_DATABASE__SHOW_ALL _SUBCLASSES	Show All Subclasses Of -- The name of the Database_showallsubclassesof operation

---

---

11143	VUTIL_DATABASE_SHOW_ALL _SUBCLASSES__CLASS	Subclasses Of : Arg 1 -- the class
11144	VUTIL_DATABASE__BAD _INTERPERSONAL_SYNC	Copying classes between personal databases is not supported. Can't synclass between personals since you can't have two personals in the same session.
11145	VUTIL_DATABASE__COULD_NOT _REFRESH_CLASSES	Error refreshing database's classes list. Error given when Database refreshclasses fails
11146	VUTIL_DATABASE__USER_NOT_IN _GROUP	User %s is not in group %s -- Error for attempt to add a user to a group database that is not in the appropriate group.
11147	VUTIL_DATABASE__BEGIN_SESSION _ERROR	Error while beginning session with personal database %s -- error for failed begin session.
11148	VUTIL_DATABASE__CONNECTDB _ERROR	Error while connecting to database %s.error for failed connectdb.
11149	VUTIL_DATABASE__END_SESSION _ERROR	Error while ending session with personal database %s. Error for failed end session.
11150	VUTIL_DATABASE__DISCONNECT _ERROR	Error while beginning session with database %s. Error for failed disconnectdb.
11151	VUTIL_DATABASE__BACKUPINFO _FORMAT_SANS_COMMENT	Level %d made on %s. The format for listing a backup info entry w/o comment
11152	VUTIL_DATABASE__BACKUPINFO _FORMAT_WITH_COMMENT	Level %d made on %s. %s The format for listing a backup info entry + comment
11153	VUTIL_DATABASE__COULD_NOT _REFRESH_BACKUP_INFO	Error refreshing database's backup information. Error for when refreshBackupInfo fails
11154	VUTIL_DATABASE__BACKUP	Online Incremental Backup -- name of database_backup operation.

11155	VUTIL_DATABASE__ADD_GROUP	Add Group -- name of database_addgroup operation
11156	VUTIL_DATABASE_ADD_GROUP__GROUP	Group to add -- group arg for database_addgroup operation
11157	VUTIL_DATABASE__REMOVE_GROUP	Remove Group -- name of database_removegroup operation
11158	VUTIL_DATABASE_REMOVE_GROUP__GROUP	Group to remove -- group arg for database_removegroup operation
11159	VUTIL_DATABASE_ADD_USER__USER	User to add -- user arg for database_adduser operation
11160	VUTIL_DATABASE__COULD_NOT_ADD	Could not add user %s -- error when adduser() fails.
11161	VUTIL_DATABASE__COULD_NOT_REMOVE	Could not remove user %s -- error when removeuser() fails.
11162	VUTIL_DATABASE__MODE_UNSTARTABLE	Unstartable -- name of the unstartable mode.
11163	VUTIL_DATABASE__MODE_SINGLE_CONNECT	DBA Only; Single Connection -- name of the single process mode.
11164	VUTIL_DATABASE__MODE_DBA_ONLY	DBA Only; Multiple Connections -- name of the dba only mode.
11165	VUTIL_DATABASE__MODE_MULTI_USER	Multi-User -- name of the started mode.
11166	VUTIL_DATABASE__FORCE_MESSAGE	Database %s is in use. Is it ok\nto kill the processes using it? -- Question asked before "forcing" a change in mode.
11167	VUTIL_DATABASE__COULD_NOT_REFRESH_SIZE	Error refreshing database's size -- When Database::refreshsize fails
11168	VUTIL_DATABASE__COULD_NOT_REFRESH_MODE	Error refreshing database's mode -- When Database::refreshmode fails
11169	VUTIL_DATABASE__COULD_NOT_REFRESH_USERS	Error refreshing database's user list -- When Database::refre-shusers fails



---

11170	VUTIL_DATABASE__UNSTARTABLE__SKIP	Database in Unstartable mode -- skipping refresh of users classes, and size. Message for when refreshes are skipped due to unstartable mode
11171	VUTIL_DATABASE_SCH2DB__DESCRIPTION	Enter the .sch file name(s) and pick one of the three\nstrategies for resolving conflicts with existing\nclasses : Abort the operation, Drop the existing\nclasses, or Evolve the existing classes. -- The message displayed in the dialog box
11172	VUTIL_DATABASE__ROLL_FORWARD_ON	Turn On Log Archiving -- name of database_rollforwardon operation
11173	VUTIL_DATABASE__ROLL_FORWARD_OFF	Turn Off Log Archiving -- name of database_rollforwardon operation
11174	VUTIL_DATABASE__ROLL_FORWARD_LOG	Archive Logs -- name of database_rollforwardon operation
11176	VUTIL_DATABASE_VIEW__CLASSNAME	Database View -- The name of the Database View class
11177	VUTIL_DATABASE_VIEW__VERSION_NAME	Version -- The name of the Version field
11178	VUTIL_DATABASE_VIEW__CREATIONDATE_NAME	Creation Date -- The name of the Creation Date field
11179	VUTIL_DATABASE_VIEW__ISPUBLIC_NAME	Disregard User List -- The name of the Public Access field
11180	VUTIL_DATABASE_VIEW__USERS_NAME	Users -- The name of the Users field
11181	VUTIL_DATABASE_VIEW__CLASSES_NAME	Classes -- The name of the Classes field
11182	VUTIL_DATABASE_VIEW__VERSION_LABEL	Version -- The label of the Version field
11183	VUTIL_DATABASE_VIEW__CREATIONDATE_LABEL	Created -- The label of the Creation Date field
11184	VUTIL_DATABASE_VIEW__ISPUBLIC_LABEL	Disregard User List -- The label of the Public Access field

---

---

11185	VUTIL_DATABASE_VIEW__USERS _LABEL	Users -- The label of the Users field
11186	VUTIL_DATABASE_VIEW__CLASSES _LABEL	Classes -- The label of the Classes field
11187	VUTIL_DATABASE_VIEW__WINDOW _TITLE	Versant Utilities; Database %s - - The title of the Database Window
11188	VUTIL_DATABASE_VIEW __BACKUPINFO_NAME	Backups -- The name of the backupinfo field
11189	VUTIL_DATABASE_VIEW __BACKUPINFO_LABEL	Current Incremental Backups -- The label for the backupinfo field
11190	VUTIL_DATABASE_VIEW__MODE _NAME	Mode -- The name of the mode field
11191	VUTIL_DATABASE_VIEW__MODE _LABEL	Mode -- The label for the mode field
11192	VUTIL_DATABASE_VIEW__SIZE _NAME	Size -- The name of the size field
11193	VUTIL_DATABASE_VIEW__SIZE _LABEL	Size -- The label for the size field
11194	VUTIL_DATABASE_VIEW__FREE _NAME	Free -- The name of the free field
11195	VUTIL_DATABASE_VIEW__FREE _LABEL	Free -- The label for the free field
11201	VUTIL_DATABASE_DIRECTORY __CLASSNAME	Database Directory -- The name of the Database Directory class
11202	VUTIL_DATABASE_DIRECTORY __PERSONAL	Personal -- The database type name of personal databases
11203	VUTIL_DATABASE_DIRECTORY __GROUP	Group -- The database type name of group databases
11204	VUTIL_DATABASE_DIRECTORY __UNIVERSAL	Universal -- The database type name of universal databases

---

11205	VUTIL_DATABASE_DIRECTORY __NEW_DB_NOT_FOUND	Could not find newly created database directory - error for when create succeeds but refresh fails
11206	VUTIL_DATABASE_DIRECTORY __DESTROY	Remove -- name of the databasedirectory_destroy operation
11207	VUTIL_DATABASE_DIRECTORY __CREATEDB	Create Database -- name of the databasedirectory_createdatabase operation
11208	VUTIL_DATABASE_DIRECTORY __CREATEDB__OPTIONS	Initialize the database : Arg 1 - options for createdb init or don't init
11209	VUTIL_DATABASE_DIRECTORY __CREATEDB__INIT	Reserve : Arg 1; option 1 -- init
11210	VUTIL_DATABASE_DIRECTORY __CREATEDB__DONT_INIT	Grow : Arg 1; option 2 -- dont't init
11211	VUTIL_DATABASE_DIRECTORY __CREATEDB__MESSAGE	Should disk space be reserved for the full size of the system volume , or should it be\ncreated at a minimal size and allowed to dynamically grow up to its maximum size? -- The message displayed in the operation dialog box
11212	VUTIL_DATABASE_DIRECTORY __REFRESH	Refresh -- The name of the databasedirectory_refresh operation
11213	VUTIL_DATABASE_DIRECTORY __PARAMETERS	View Parameters -- The name of the databasedirectory_parameters operation
11214	VUTIL_DATABASE_DIRECTORY__NOT_ROOT	Only the superuser can change the owner -- Error for when the owner is changed by other than root
11215	VUTIL_DATABASE_DIRECTORY __VIEW	View -- The name of the databasedirectory_view operation

---

11216	VUTIL_DATABASE_DIRECTORY__NO __LONGER_EXISTS	%s no longer exists. -- The mes- sage given when a database or database directory is destroyed.
11217	VUTIL_DATABASE_DIRECTORY __NAME_IN_USE	The name %s is already in use -- Given when makedb is called with an already used name.
11218	VUTIL_DATABASE_DIRECTORY __RESTORE	Restore from Backup -- name of database_restore operation.
11251	VUTIL_DATABASE_DIRECTORY_VIEW __CLASSNAME	Database Directory View -- The name of the Database Directory View class
11252	VUTIL_DATABASE_DIRECTORY_VIEW __NAME_NAME	Name -- The name of the Name field
11253	VUTIL_DATABASE_DIRECTORY_VIEW __OWNER_NAME	DBA -- The name of the Owner field
11254	VUTIL_DATABASE_DIRECTORY_VIEW __GROUP_NAME	Group -- The name of the Group field
11255	VUTIL_DATABASE_DIRECTORY_VIEW __DBTYPE_NAME	Database Type -- The name of the Database Type field
11256	VUTIL_DATABASE_DIRECTORY_VIEW __LOGGING_NAME	Logging -- The name of the Log- ging field
11258	VUTIL_DATABASE_DIRECTORY_VIEW __LOCKING_NAME	Locking -- The name of the Lock- ing field
11260	VUTIL_DATABASE_DIRECTORY_VIEW __NAME_LABEL	Name -- The label of the Name field
11261	VUTIL_DATABASE_DIRECTORY_VIEW __OWNER_LABEL	DBA -- The label of the Owner field
11262	VUTIL_DATABASE_DIRECTORY_VIEW __GROUP_LABEL	Group -- The label of the Group field
11263	VUTIL_DATABASE_DIRECTORY_VIEW __DBTYPE_LABEL	Type -- The label of the Database Type field
11264	VUTIL_DATABASE_DIRECTORY_VIEW __LOGGING_LABEL	Logging -- The label of the Log- ging field
11266	VUTIL_DATABASE_DIRECTORY_VIEW __LOCKING_LABEL	Locking -- The label of the Lock- ing field

---

11268	VUTIL_DATABASE_DIRECTORY_VIEW __WINDOW_TITLE	Versant Utilities; Database Directory %s -- The title of the Database Directory Window
11301	VUTIL_FRONTEND_PROFILE __CLASSNAME	Application Process Parameters - - The name of the frontendpro- file class
11302	VUTIL_ALIAS__CLASSNAME	Alias -- The name of the Alias class
11303	VUTIL_FRONTEND_PROFILE__ADD _CONNECTION	Add -- The name of the frontendprofile_addconnection operation
11304	VUTIL_FRONTEND_PROFILE_ADD _CONNECTION__DATABASE	Database -- The first arg -- the name of the database to connect to
11305	VUTIL_FRONTEND_PROFILE __REMOVE_CONNECTION	Remove Selection -- The name of the frontendprofile_removeconnection operation
11306	VUTIL_FRONTEND_PROFILE__ADD _ALIAS	Add -- The name of the frontendprofile_addalias opera- tion
11307	VUTIL_FRONTEND_PROFILE_ADD _ALIAS__ALIAS	Alias -- The first arg -- the alias
11308	VUTIL_FRONTEND_PROFILE_ADD _ALIAS__DATABASE	Database -- The second arg -- the expansion
11309	VUTIL_FRONTEND_PROFILE __REMOVE_ALIAS	Remove Selection -- The name of the frontendprofile_removealias operation
11310	VUTIL_FRONTEND_PROFILE__BAD _DATABASE_NAME	Bad database name %s -- A data- base name in the connects or aliases list was bad
11311	VUTIL_FRONTEND_PROFILE__BAD _CONNECT	Bad entries removed from con- nect list -- There were bad data- base names in the connects list
11312	VUTIL_FRONTEND_PROFILE__BAD _ALIAS	Bad entries removed from alias list -- There were bad database names in the aliases list

11313	VUTIL_FRONTEND_PROFILE __PERSONAL_CONNECT	You cannot have an automatic connection to a personal database -- user tries to add a personal database to the connects list
11314	VUTIL_FRONTEND_PROFILE __UNCREATED_CONNECT	Be careful it is impossible to connect to an uncreated database directory -- The user added a connection to an uncreated database
11315	VUTIL_FRONTEND_PROFILE__READ __FAILED	Error reading Client Process parameter file -- error when read fails
11316	VUTIL_FRONTEND_PROFILE__WRITE __FAILED	Error writing Client Process parameter file -- error when write fails
11351	VUTIL_GROUP__CLASSNAME	Group -- The name of the Group class
11401	VUTIL_MACHINE__CLASSNAME	Host -- The name of the Machine class
11402	VUTIL_MACHINE__REFRESH	Refresh -- The name of the Machine_refresh operation
11403	VUTIL_MACHINE__OPEN_MACHINE	New Host Window -- The name of the Machine_newmachine operation
11404	VUTIL_MACHINE_OPEN_MACHINE __MACHINE_NAME	Name of Host : Arg 1 -- The name of the machine to open
11405	VUTIL_MACHINE__VIEW	View Selection -- The name of the Machine_view operation
11406	VUTIL_MACHINE__NEW_DIRECTORY	New Database Directory -- The name of the Machine_createdirectory operation
11407	VUTIL_MACHINE_NEW_DIRECTORY __DIR_NAME	Name of the new directory : Arg 1 -- the name of the new directory
11408	VUTIL_MACHINE_NEW_DIRECTORY __INVALID_NAME	Invalid host name -- Error generated when the machine name is invalid.

---

11409	VUTIL_MACHINE__CREATE _DATABASES	Create Database(s) from Selection -- The name of the Machine_createdatabases operation
11410	VUTIL_MACHINE_CREATE _DATABASES__OPTIONS	Initialize the database : Arg 1 - - options for createdb: init or don't init
11411	VUTIL_MACHINE_CREATE _DATABASES__INIT	Reserve : Arg 1, option 1 -- init
11412	VUTIL_MACHINE_CREATE _DATABASES__DONT_INIT	Grow : Arg 1, option 2 -- don't init
11413	VUTIL_MACHINE_CREATE _DATABASES__MESSAGE	Should disk space be reserved for the full size of the system volume , or should it be created at a minimal size and allowed to dynamically grow up to its maximum size?
11414	VUTIL_MACHINE__DESTROY _DIRECTORIES	Remove Selection -- The name of the Machine_destroydirectories operation
11415	VUTIL_MACHINE_DESTROY _DIRECTORIES__MESSAGE	Remove the following database directories? -- The message dis- played in the operation dialog box
11416	VUTIL_MACHINE__START	Start Selection -- The name of the Machine_startdbs operation
11417	VUTIL_MACHINE__STOP	Stop Selection -- The name of the Machine_stopdbs operation
11418	VUTIL_MACHINE_STOP__OPTIONS	Force to stop : Arg 1 -- options for stopdb: force or don't force
11419	VUTIL_MACHINE_STOP__FORCE	Force : Arg 1: option 1 -- force
11420	VUTIL_MACHINE_STOP__DONT _FORCE	Don't Force : Arg 1: option 2 -- don't force
11421	VUTIL_MACHINE_STOP__MESSAGE	Force killing of all active transactions? -- The message displayed in the operation dia- log box

---

11422	VUTIL_MACHINE__DESTROY _DATABASES	Remove Selection -- The name of the Machine_destroydatabases operation
11423	VUTIL_MACHINE_DESTROY _DATABASES__OPTIONS	Force to remove : Arg 1 : options for removedb; force or don't force
11424	VUTIL_MACHINE_DESTROY _DATABASES__FORCE	Force : Arg 1 : option 1 -- force
11425	VUTIL_MACHINE_DESTROY _DATABASES__DONT_FORCE	Don't Force : Arg 1 : option 2 - - dont't force
11426	VUTIL_MACHINE_DESTROY _DATABASES__MESSAGE	Destroy all objects in the following databases? -- The message displayed in the operation dialog box
11427	VUTIL_MACHINE__WRONG_PERSONAL	This personal database is only accessible by its owner: %s -- The message displayed when a user tries to access someone else's personal database.
11428	VUTIL_MACHINE__NOT_IN_GROUP	To access this group database you must be in the %s group -- The message displayed when a user tries to access a group database for another group.
11429	VUTIL_MACHINE__NOT_IN_USER _LIST	To access this database you must be in its user list -- The message displayed when a user is not in a database's user list.
11430	VUTIL_MACHINE__BACKUPDBS	Online Incremental Backup of Selection -- The name of the machine_backupdbs operation.
11431	VUTIL_MACHINE__RESTOREDBS	Restore Selection from Backup -- The name of the machine_restoredbs operation.
11432	VUTIL_MACHINE__CREATE_DEVICE	New Backup Device Window -- name of database_createdevice operation.



---

11433	VUTIL_MACHINE__VIEW _PARAMETERS	View Parameters of Selection -- name of machine_viewparameters operation
11434	VUTIL_MACHINE__ROLL_FORWARD _ON	Turn On Log Archiving for Selec- tion -- name of machine_rollforwardon operation
11435	VUTIL_MACHINE__ROLL_FORWARD _OFF	Turn Off Log Archiving for Selection -- name of machine_rollforwardon operation
11436	VUTIL_MACHINE__ROLL_FORWARD _LOG	Archive Logs for Selection -- name of machine_rollforwardon operation
11451	VUTIL_MACHINE_VIEW__CLASSNAME	Host View -- The name of the Machine View class
11452	VUTIL_MACHINE_VIEW__WINDOW _TITLE	Versant Utilities: Host %s -- the title of the Machine Window
11453	VUTIL_MACHINE_VIEW__DATABASE _LIST_NAME	Databases -- the name of the database list
11454	VUTIL_MACHINE_VIEW __DIRECTORIES_LIST_NAME	Database Directories -- the name of the database directories list
11455	VUTIL_MACHINE_VIEW__ROOT_PATH _NAME	Root Path -- the name of the root path field
11456	VUTIL_MACHINE_VIEW__DB_PATH _NAME	Database Path -- the name of the db path field
11457	VUTIL_MACHINE_VIEW__VERSION _NAME	Product Version -- the name of the product version field
11458	VUTIL_MACHINE_VIEW__DATABASE _LIST_LABEL	Fully Created Databases -- the label of the database list
11459	VUTIL_MACHINE_VIEW __DIRECTORIES_LIST_LABEL	Empty Database Directories -- the label of the database direc- tories list
11460	VUTIL_MACHINE_VIEW__ROOT_PATH _LABEL	Root Path -- the label of the root path field
11461	VUTIL_MACHINE_VIEW__DB_PATH _LABEL	Database Path -- the label of the db path field
11462	VUTIL_MACHINE_VIEW__VERSION _LABEL	Product Version -- the label of the product version field

11463	VUTIL_MACHINE_VIEW__CLOSE	Close -- The name of the machineview_close window
11464	VUTIL_MACHINE_VIEW__ARE_YOU__SURE	Are you sure you want to exit Versant Utilities? -- The question to ask the user before exiting
11465	VUTIL_MACHINE_VIEW__DBID_NODE__NAME	DBID Node -- the name of the dbid node field
11466	VUTIL_MACHINE_VIEW__DBID_PATH__NAME	DBID Path -- the name of the dbid path field
11467	VUTIL_MACHINE_VIEW__DBID_NODE__LABEL	DBID Node -- the label of the dbid node field
11468	VUTIL_MACHINE_VIEW__DBID_PATH__LABEL	DBID Path -- the name of the dbid path field
11501	VUTIL_MAIN__PERSONAL_DB_FLAG__ALIAS	-D -- A localized alias for -D
11502	VUTIL_MAIN__NO_ARG_ERROR	-D requires an argument -- printed when no arg is supplied to -D
11503	VUTIL_MAIN__IGNORING_FLAG	Ignoring unknown %s flag -- printed when an unknown flag is found
11504	VUTIL_MAIN__NOT_FOUND	Ignoring non-existent %s -- printed when a non-existent database or database directory is passed.
11505	VUTIL_MAIN__APP_NAME	Versant Utilities -- The name of the application: Versant Utilities
11506	VUTIL_MAIN__VIEWS_UNAVAILABLE	The DISPLAY environment variable must be set -- The error message to give when no X display is specified
11507	VUTIL_MAIN__COULD_NOT_REFRESH__LOCAL_HOST	Error refreshing information on the local host -- Error given if refresh fails for the local host

---

11508	VUTIL_MACHINE__COULD_NOT_REFRESH_OSCP_INFO	Error refreshing host's configuration information -- error given of Machine::refreshinfo fails
11509	VUTIL_MACHINE__COULD_NOT_REFRESH_DBLIST_INFO	Error refreshing host's database list and related information -- error given of Machine::refreshdatabases fails
11601	VUTIL_PROFILE_VIEW__CLASSNAME	Parameters View -- The name of the profileview class
11602	VUTIL_PROFILE_VIEW__SYSVOL_LABEL	System Volume -- The label of the system volume profile parameter
11603	VUTIL_PROFILE_VIEW__PLOGVOL_LABEL	Physical Log Volume -- the label of the Physical Log Volume profile parameter
11604	VUTIL_PROFILE_VIEW__LLOGVOL_LABEL	Logical Log Volume -- the label of the Logical Log Volume profile parameter
11605	VUTIL_PROFILE_VIEW__VOL_PATH_LABEL	Path -- the label of the Path profile parameter
11606	VUTIL_PROFILE_VIEW__VOL_SIZE_LABEL	Size -- the label of the Size profile parameter
11607	VUTIL_PROFILE_VIEW__EXTENT_SIZE_LABEL	Pages per Extent -- the label of the Pages per Extent profile parameter
11608	VUTIL_PROFILE_VIEW__LLOG_BUF_SIZE_LABEL	Logical Log Buffer Size -- the label of the Logical Log Buffer Size profile parameter
11609	VUTIL_PROFILE_VIEW__PLOG_BUF_SIZE_LABEL	Physical Log Buffer Size -- the label of the Physical Log Buffer Size profile parameter
11610	VUTIL_PROFILE_VIEW__HEAP_SIZE_LABEL	Initial Heap Size -- the label of the Initial Heap Size profile parameter
11611	VUTIL_PROFILE_VIEW__MAX_PAGE_BUFFS_LABEL	Maximum Page Buffers -- the label of the Maximum Page Buffers profile parameter

---

11612	VUTIL_PROFILE_VIEW__LOCK_WAIT _TIMEOUT_LABEL	Lock Wait Timeout -- the label of the Lock Wait Timeout profile parameter
11613	VUTIL_PROFILE_VIEW__DATABASE _TIMEOUT_LABEL	Database Timeout -- the label of the Database Timeout profile parameter
11614	VUTIL_PROFILE_VIEW__CREATION _LABEL	Creation Parameters -- the label of the Creation Parameters group label
11615	VUTIL_PROFILE_VIEW __BACKEND_LABEL	Server Process Tuning Parameters -- the label of the Server Process Parameters group label
11616	VUTIL_PROFILE_VIEW__FRONTEND _LABEL	Application Process Parameters for %s -- the label of the Application Process Parameters group label
11617	VUTIL_PROFILE_VIEW__CACHE _SIZE_LABEL	Swap Threshold -- the label of the Swap Threshold profile parameter
11618	VUTIL_PROFILE_VIEW__FE_HEAP _SIZE_LABEL	Initial Heap Size -- the label of the Initial Heap Size profile parameter
11619	VUTIL_PROFILE_VIEW__ALIASES _LABEL	Aliases -- the label of the Aliases profile parameter
11620	VUTIL_PROFILE_VIEW__CONNECTS _LABEL	Automatic Connections -- the label of the Automatic Connections profile parameter
11621	VUTIL_PROFILE_VIEW__WINDOW _TITLE	Versant Utilities: Parameters for %s -- the title of the Parameters Window
11623	VUTIL_PROFILE_VIEW__SYSVOL _PATH_NAME	System Volume Path -- the name of the System Volume Path profile parameter
11624	VUTIL_PROFILE_VIEW__PLOGVOL _PATH_NAME	Physical Log Volume Path -- the name of the Physical Log Volume Path profile parameter
11625	VUTIL_PROFILE_VIEW__LLOGVOL _PATH_NAME	Logical Log Volume Path -- the name of the Logical Log Volume Path profile parameter

---

11626	VUTIL_PROFILE_VIEW__EXTENT _SIZE_NAME	Pages per Extent -- the name of the Pages per Extent profile parameter
11627	VUTIL_PROFILE_VIEW__SYSVOL _SIZE_NAME	System Size -- the name of the System Size profile parameter
11628	VUTIL_PROFILE_VIEW__LLOGVOL _SIZE_NAME	Logical Log Size -- the name of the Logical Log Size profile parameter
11629	VUTIL_PROFILE_VIEW__PLOGVOL _SIZE_NAME	Physical Log Size -- the name of the Physical Log Size profile parameter
11630	VUTIL_PROFILE_VIEW__LLOG_BUF _SIZE_NAME	Logical Log Buffer Size -- the name of the Logical Log Buffer Size profile parameter
11631	VUTIL_PROFILE_VIEW__PLOG_BUF _SIZE_NAME	Physical Log Buffer Size -- the name of the Physical Log Buffer Size profile parameter
11632	VUTIL_PROFILE_VIEW__HEAP_SIZE _NAME	Server Process Initial Heap Size -- the name of the Server Process Initial Heap Size profile parameter
11633	VUTIL_PROFILE_VIEW__MAX_PAGE _BUFFS_NAME	Maximum Page Buffers -- the name of the Maximum Page Buffers profile parameter
11634	VUTIL_PROFILE_VIEW__LOCK_WAIT _TIMEOUT_NAME	Lock Wait Timeout -- the name of the Lock Wait Timeout profile parameter
11635	VUTIL_PROFILE_VIEW__DATABASE _TIMEOUT_NAME	Database Timeout -- the name of the Database Timeout profile parameter
11636	VUTIL_PROFILE_VIEW__CACHE _SIZE_NAME	Swap Threshold -- the name of the Swap Threshold profile parameter
11637	VUTIL_PROFILE_VIEW__FE_HEAP _SIZE_NAME	Application Process Initial Heap Size -- the name of the Application Process Initial Heap Size profile parameter
11638	VUTIL_PROFILE_VIEW__ALIASES _NAME	Aliases -- the name of the Aliases profile parameter

11639	VUTIL_PROFILE_VIEW__CONNECTS __NAME	Automatic Connections -- the name of the Automatic Connections profile parameter
11640	VUTIL_PROFILE_VIEW __TRANSACTIONS_LABEL	Transaction Count -- the label of the transactions profile parameter
11641	VUTIL_PROFILE_VIEW __TRANSACTIONS_NAME	Transaction Count -- the name of the transactions profile parameter
11642	VUTIL_PROFILE_VIEW__MAX __PROCESSES_LABEL	Maximum Processes -- the label of the max_processes profile parameter
11643	VUTIL_PROFILE_VIEW__MAX __PROCESSES_NAME	Maximum Processes -- the name of the max_processes profile parameter
11644	VUTIL_PROFILE_VIEW__COMMIT __FLUSH_LABEL	Commit Flush -- the label of the commit_flush profile parameter
11645	VUTIL_PROFILE_VIEW__COMMIT __FLUSH_NAME	Commit Flush -- the name of the commit_flush profile parameter
11646	VUTIL_PROFILE_VIEW__SIGNAL __BLOCK_LABEL	Signal Block -- the label of the signal_block profile parameter
11647	VUTIL_PROFILE_VIEW__SIGNAL __BLOCK_NAME	Signal Block -- the name of the signal_block profile parameter
11651	VUTIL_USER__CLASSNAME	User -- The name of the User class
11701	VUTIL_CLASS__CLASSNAME	Class -- The name of the Class class
11751	VUTIL_DEVICE__CLASSNAME	Backup Device -- The name of the Device class
11752	VUTIL_DEVICE__LEVEL_ARG	Level -- The label for the level argument
11753	VUTIL_DEVICE__POSITION_ARG	Tape Position -- The label for the level argument
11754	VUTIL_DEVICE__COMMENT_ARG	Comment -- The label for the level argument
11755	VUTIL_DEVICE__DEVICE_ARG	Device -- The label for the device argument

---

11756	VUTIL_DEVICE__BLOCKING_ARG	Blocking -- The label for the blocking argument
11757	VUTIL_DEVICE__CAPACITY_ARG	Capacity -- The label for the capacity argument
11758	VUTIL_DEVICE__DATABASE_ARG	Database -- The label for the database argument
11759	VUTIL_DEVICE__BACKUP	Online Incremental Backup -- The name of device_backup
11760	VUTIL_DEVICE__REFRESH	Refresh -- The name of device_refresh
11761	VUTIL_DEVICE__MOUNT_THE_NEXT_TAPE	Mount the appropriate tape... -- printed between restores
11762	VUTIL_DEVICE__ANOTHER_LEVEL_QUERY	Would you like to do another level of restore\nnon database %s? -- asks the user if they would like to do another level of restore
11763	VUTIL_DEVICE__MODAL_QUERY	Modal Query -- The name of device_recoverableerror
11764	VUTIL_DEVICE__BACKUP_MESSAGE	Would you like to turn on log archiving\nwith this backup? Prepare the backup device\nand press 'Yes' or 'No' to start the backup. -- The message in the backup dialog
11765	VUTIL_DEVICE__RESTORE_MESSAE	Prepare the backup device and\npress 'OK' to start the restore. -- The message in the restore dialog
11766	VUTIL_DEVICE__LOG_MESSAGE	Prepare the backup device and press\n'ok' to start archiving log records. -- The message in the backup dialog
11767	VUTIL_DEVICE__ROLL_FORWARD_LOG	Archive Logs -- name of machine_rollforwardlog operation
11768	VUTIL_DEVICE__AGGRESSION_ARG	Aggression -- The label for the database argument

---

11769	VUTIL_DEVICE__LOG_RESTORE _MESSAGE	Reappling log records... -- Mes- sage displayed during log roll forward restore
11770	VUTIL_DEVICE__COPY_QUERY	During roll forward would you like to apply records\nfrom the database's current log file in addition to any\narchived records ? If so, then this log file must be\ncopied to a safe location so it is not overwrit- ten\nduring the restore.
11771	VUTIL_DEVICE__OLD_LOG_ARG	Path of log file
11772	VUTIL_DEVICE__COPY_LOG_ARG	Path for copy of log file
11773	VUTIL_DEVICE__RF_ON	Yes
11774	VUTIL_DEVICE__RF_UNCHANGED	No
11775	VUTIL_DEVICE__RF_ARG	Turn on roll forward
11786	VUTIL_DEVICE_VIEW__CLASSNAME	Backup Device View -- The name of the deviceview class
11787	VUTIL_DEVICE_VIEW__WINDOW _TITLE	Versant Utilities; Backup Device %s -- the title of the Device window
11788	VUTIL_DEVICE_VIEW__FILES _LABEL	Backup files on device -- The text of the backup files label
11789	VUTIL_DEVICE_VIEW__FILES_NAME	Files -- The name of the backup files list
11790	VUTIL_DEVICE_VIEW__CAPACITY _LABEL	Capacity -- The text of the capacity label
11791	VUTIL_DEVICE_VIEW__CAPACITY _NAME	Capacity -- The name of capacity field
11792	VUTIL_DEVICE_VIEW__BLOCKING _LABEL	Blocking -- The text of the blocking label
11793	VUTIL_DEVICE_VIEW__BLOCKING _NAME	Blocking -- The name of blocking field
11794	VUTIL_DEVICE_VIEW__DEVICE _LABEL	Device -- The text of the device label



---

11795	VUTIL_DEVICE_VIEW__DEVICE _NAME	Device -- The name of device field
11801	VUTIL_FILE__CLASSNAME	Backup Device View -- The name of the deviceview class
11802	VUTIL_FILE__NAME_FORMAT_WITH _COMMENT	Position %d: %s volume %d. %s - - The format of a file's name
11803	VUTIL_FILE__NAME_FORMAT_SANS _COMMENT	Position %d: %s volume %d -- The format of a file's name
11804	VUTIL_FILE__RESTORE	Restore Database from Selection -- The name of file_restore
11805	VUTIL_FILE_RESTORE__DATABASE _ARG	Database -- The database arg for file_restore
11806	VUTIL_FILE_RESTORE__SINGLE _FILE_ONLY	Multiple file restores are not yet supported.\nselect a single file and try again. -- Error given when multi-file select is attempted.
12301	VBACKUP_ALGO__NEXT_TAPE	End of tape encountered. Please mount a blank tape. VBACKUP_TAPE__END_OF_VOLUME dur- ing restore
12302	VBACKUP_ALGO__COULD_NOT_OPEN	Could not open device %s at posi- tion %s.\nos error %#d. Please check the device. -- verbose version of VBACKUP_TAPE__COULD_NOT_OPEN
12303	VBACKUP_ALGO__WRONG_FORMAT	The file on device %s at position %s is not\nof the proper format. Please mount the proper tape. -- verbose version of VBACKUP_TAPE__WRONG_FORMAT
12304	VBACKUP_ALGO__WRONG_VOLUME	The file on device %s at position %s\nis volume %d. Please mount volume %d. -- verbose version of VBACKUP_TAPE__WRONG_VOLUME

12305	VBACKUP_ALGO__WRONG_LABEL	The file on device %s at position %s is labeled :\n %s\nplease retry with the proper file:\n %s -- verbose version of VBACKUP_TAPE__WRONG_LABEL
12322	VBACKUP_TAPE__END_OF_VOLUME	Please mount the next tape -- End of volume reached.
12323	VBACKUP_TAPE__COULD_NOT_OPEN	Error while opening the file -- An error was encountered opening the file
12324	VBACKUP_TAPE__WRONG_FORMAT	Wrong file format -- The file format was different from the one specified in open()
12325	VBACKUP_TAPE__WRONG_LABEL	Wrong file label -- The label was different from the one specified in open()
12326	VBACKUP_TAPE__WRONG_VOLUME	Wrong volume number -- The tapes were inserted in the wrong order for a multi volume file.
12327	VBACKUP_TAPE__WRONG_BLOCK	Block check failure. File is corrupted -- The internal block field was wrong indicating file corruption.
12328	VBACKUP_TAPE__ABORTED	Aborted at user's request -- the backup was aborted.
12329	VBACKUP_TAPE__ERROR	Error reading or writing the tape -- error from read(), write(), or rmt connection.
12330	VBACKUP_TAPE__GET_STATUS_FAILED	Failed ioctl call while getting status of device.
12331	VBACKUP_TAPE__EOM_FAILED	Failed ioctl call while positioning tape after last file.
12332	VBACKUP_TAPE__BSF_FAILED	Failed ioctl call while adjusting the tape position backward.
12333	VBACKUP_TAPE__FSF_FAILED	Failed ioctl call while adjusting the tape position forward.
12334	VBACKUP_TAPE__INVALID_POSITION	Can not go to a negative tape position.

---

12335	VBACKUP_TAPE__EOF_FAILED	Failed ioctl call while writing end-of-file marker.
12336	VBACKUP_TAPE__CLOSE_FAILED	Could not close the device. OS error #&d.
12337	VBACKUP_TAPE__BROKEN_PIPE	The pipe connecting to the remote device was broken.
12338	VBACKUP_TAPE__NO_ACCESS_TO_HOST	Permission denied while trying to access remote host via rsh.
12339	VBACKUP_TAPE__PIPE_CREATION_FAILED	Error creating a pipe.
12340	VBACKUP_TAPE__FORK_FAILED	Could not fork the rsh process probably due to lack of swap space.
12341	VBACKUP_TAPE__COULD_NOT_CLOSE_PIPE	Error while closing pipe to remote tape device.
12342	VBACKUP_TAPE__SEND_COMMAND_FAILED	Error while sending a command to the remote device.
12343	VBACKUP_TAPE__POSITION_0_ONLY	Files only support position 0.
12344	VBACKUP_TAPE__REWIND_FAILED	Failed ioctl call while rewinding the tape.
12345	VBACKUP_TAPE__FLUSH_FAILED	Error while syncing backup device.
12346	VBACKUP_TAPE__WRONG_VERSION	The backup was made with an old incompatible version of Versant.
12347	VBACKUP_TAPE__WRONG_BYTE_ORDERING	The backup was made on a machine with incompatible byte ordering.
12348	VBACKUP_TAPE__REWINDING	Warning: %s is a rewinding tape device.\nonly non-rewinding tape devices are supported.
12349	VBACKUP_TAPE__DBUFFER_FULL	Double buffer used for backup and restore is currently full cannot copy into full buffers
12350	VBACKUP_TAPE__DBUFFER_EMPTY	Double buffer used for backup and restore is currently empty cannot "get" from empty buffers
12351	VBACKUP_OLIB__UNKNOWN_TIMESTAMP	??? Given as timestamp when backupdbinfo fails

---

12352	VBACKUP_OLIB__NEXT_BACKUP _TAPE	End of tape encountered. Please mount a blank\ntape on device %s. VBACKUP_TAPE__END_OF_VOLUME during backup
12353	VBACKUP_OLIB__NEXT_RESTORE _TAPE	End of tape encountered. Please mount volume %d\non device %s. VBACKUP_TAPE__END_OF_VOLUME during restore
12354	VBACKUP_OLIB__WRONG_DBNAME	The file on device %s at position %s is a backup of\ndatabase %s. Please try again using a backup of %s. -- verbose version of VBACKUP_TAPE__WRONG_DBNAME
12355	VBACKUP_OLIB__INCONSISTENT _RESTORE	The database %s @ %s should be restored with the\nnew name %s @ %s. Please try again. -- verbose version of VBACKUP_TAPE__INCONSISTENT_RESTORE
12359	VBACKUP_OLIB__NEED_PREVIOUS _LEVEL	Before you can restore this level %d backup,\nyou must first restore the level %d backup\nmade on %s -- verbose version of UT_ER_NEED_LEVEL_XXX errors.
12360	VBACKUP_OLIB__NOT_LATEST	Warning: A more recent level %d backup of %s was made\non %s -- verbose version of UT_W_NOT_LATEST
12361	VBACKUP_OLIB__NEWER_LEVEL_2	Warning: After this restore is complete, you will\nprobably want to restore the level 2 backup made\non %s -- verbose version of UT_W_NEWER_LEVEL_2
12362	VBACKUP_OLIB__DB_MODIFIED	Warning: The database has been modified since you did\nthe previous level of restore. You should consider\nrestarting from level 0. -- verbose version of UT_W_MODIFIED_SINCE_RESTORE

---

12363	VBACKUP_OLIB__REMOTE_DB _UNSUPPORTED	Backup and restore of remote databases is currently unsupported -- given when an attempt is made to backup or restore a remote database
12364	VBACKUP_OLIB__BEGIN	<done with pre-restore> gets passed to callback, but shouldn't be printed
12375	VBACKUP_ARGS__INVALID_UPDATE	Valid settings are: device, position, capacity, and blocking.
12376	VBACKUP_ARGS__USAGE	Usage: vbackup full usage\n This is never printed by Versant.\n - The usage message displayed when arguments aren't entered properly.
12377	VBACKUP_ARGS__MINUS_DEVICE	-device -- internationalized synonym for -device
12378	VBACKUP_ARGS__MINUS_LEVEL	-level -- internationalized synonym for -level
12379	VBACKUP_ARGS__MINUS_POSITION	-position -- internationalized synonym for -position
12380	VBACKUP_ARGS__MINUS_CAPACITY	-capacity -- internationalized synonym for -capacity
12381	VBACKUP_ARGS__MINUS_BLOCKING	-blocking -- internationalized synonym for -blocking
12382	VBACKUP_ARGS__MINUS_BUFFER	-buffer -- internationalized synonym for -buffer
12383	VBACKUP_ARGS__MINUS_COMMENT	-comment -- internationalized synonym for -comment
12384	VBACKUP_ARGS__MINUS_LIST	-list -- internationalized synonym for -list
12385	VBACKUP_ARGS__MINUS_BACKUP	-backup -- internationalized synonym for -backup
12386	VBACKUP_ARGS__MINUS_RESTORE	-restore -- internationalized synonym for -restore
12387	VBACKUP_ARGS__MINUS_INFO	-info -- internationalized synonym for -info

12395	VBACKUP_ARGS__BAD_ARG	Unrecognized option %s. -- Error message for bad args
12396	VBACKUP_ARGS__NO_COMMAND	A command is required. -- Error message for no command
12397	VBACKUP_ARGS__ONE_ARG	The %s option requires an argument. -- Error message for when no argument is given and a single argument is required
12398	VBACKUP_ARGS__ONE_OR_MORE_DBS	The %s command requires a list of one more databases -- Error message for when no argument is given and one or more arguments are required
12399	VBACKUP_ARGS__NO_ARGS	The %s command does not take any arguments -- Error message for when an argument is given when it should not be.
12400	VBACKUP_ARGS__LEVEL_USAGE	<level> must be one of: 0, 1 or 2 -- Error message for when the level argument doesn't parse
12401	VBACKUP_ARGS__POSITION_USAGE	<position> must be one of append; current; or a non-negative number -- Error message for when the position argument doesn't parse
12402	VBACKUP_ARGS__CAPACITY_USAGE	<capacity> must be a one of dynamic or a positive number of bytes -- Error message for when the capacity argument doesn't parse
12403	VBACKUP_ARGS__BLOCKING_USAGE	<blocking> must be a one of optimal or a positive number of bytes. -- Error message for when the blocking argument doesn't parse
12404	VBACKUP_ARGS__BUFFER_SIZE_USAGE	<buffer size> must be a positive number of bytes -- Error message for when the capacity argument doesn't parse

---

12410	VBACKUP_ARGS__APPEND	Append -- The symbol for setting position to LOW_LEVEL_TAPE_POSITION__APPEND
12411	VBACKUP_ARGS__CURRENT	Current -- The symbol for setting position to LOW_LEVEL_TAPE_POSITION__CURRENT
12412	VBACKUP_ARGS__OPTIMAL	Optimal -- The symbol for setting blocking to 0 so it defaults to optimal
12413	VBACKUP_ARGS__DYNAMIC	Dynamic -- The symbol for setting capacity to 0 so the hardware is trusted to detected the end of the tape.
12415	VBACKUP_ARGS__BYTES	Bytes -- The units for parsing and unparsing bytes
12416	VBACKUP_ARGS__BLOCKS	Blocks -- The units for parsing and unparsing 512 byte blocks
12417	VBACKUP_ARGS__KILOBYTES	Kilobytes -- The units for parsing and unparsing kilobytes
12418	VBACKUP_ARGS__MEGABYTES	Megabytes -- The units for parsing and unparsing megabytes
12419	VBACKUP_ARGS__GIGABYTES	Gigabytes -- The units for parsing and unparsing gigabytes
12420	VBACKUP_ARGS__MINUS _ROLLFORWARD	-rollforward -- internationalized synonym for -rollforward
12421	VBACKUP_ARGS__MINUS_OFF	-off -- internationalized synonym for -off
12422	VBACKUP_ARGS__MINUS_LOG	-log -- internationalized synonym for -log
12423	VBACKUP_ARGS__MINUS_SCRIPT	-script -- internationalized synonym for -log
12424	VBACKUP_ARGS__MINUS _AGGRESSION	-aggression -- internationalized synonym for -aggression
12425	VBACKUP_ARGS__AGGRESSION _USAGE	<aggression> must be a number of seconds >= 0 -- usage for -aggression
12426	VBACKUP_ARGS__MINUS_RENAME	-rename -- internationalized synonym for -rename

12427	VBACKUP_ARGS__USAGE_1	Usage: vbackup [options] command\n\nwhere options are one or more of:\n\n -level <level>\n -device <filename>\n -position <position>\n -capacity <capacity>\n -blocking <blocking>\n -comment <comment>\n -rollforward\n -startsync\n -aggression <aggression>\n -script <script>\n -- The usage message displayed when arguments aren't entered properly.
12428	VBACKUP_ARGS__USAGE_2	And command is one of:\n\n -backup <database-names>\n -restore { <database-name> [-rename <new-database-name>] [-overwrite]} ... \n -off <database-names>\n -log <database-names>\n -info <database-names>\n -list [-getbeprofile [-odir <path-to-save-profile>]]\n.\n\nThe usage message displayed when arguments aren't entered properly.
12429	VBACKUP_ARGS__MINUS_STARTSYNC	-startsync -- internationalized synonym for -startsy
12430	VBACKUP_ARGS__HABACKUP_USAGE1	usage: habackup [options] command\n\nwhere options are one or more of:\n\n -cmd <command>\n -level <level>\n -device <filename>\n -rollforward\n -startsync\n.\n\nThe usage message displayed when arguments aren't entered properly.



---

12432	VBACKUP_ARGS__HABACKUP_USAGE2	And command is one or more of : -split <database-names> -backup <database-names> -restore <new-database-names> [- syncwith <database-name>] -off <database-names> -log <database-names> -- The usage message displayed when argu- ments aren't entered properly.
12433	VBACKUP_ARGS__MISSING_SPLIT _CMD	-cmd <command> option must be specified -- usage for -split
12434	HABACKUP_ARGS__LEVEL_ZERO _DISABLED	Level 0 backup should be per- formed using -split -- usage for -split
12435	HABACKUP_ARGS__NO_COMMAND	A command is required -- Error message for no command
12436	VBACKUP_ARGS__MINUS_NOPROMPT	-noprompt -- internationalized synonym for -noprompt
12437	VBACKUP_ARGS__MINUS _BACKGROUND	-background -- international- ized synonym for -background
12438	HABACKUP_ARGS__MINUS_SYNCWITH	-syncwith <dbname to syncwith after restore> usage for -syn- cwith
12439	HABACKUP_OLIB__INCONSISTENT _RESTORE	The database %s should be restored with the\nnew name %s. Please try again -- verbose ver- sion of HABACKUP_TAPE__INCONSISTENT_REST ORE
12440	VBACKUP_ARGS__MINUS_OVERWRITE	-overwrite -- internationalized synonym for -overwrite
12441	VBACKUP_ARGS__MINUS _GETBEPROFILE	-getbeprofile -- internation- alized synonym for -getbeprofile
12442	VBACKUP_ARGS__MINUS_ODIR	-odir -- internationalized syn- onym for -odir

---

12450	VBACKUP_CLI__COPYRIGHT	VERSANT Utility VBACKUP Version \\%s.\\%s_--\\%s\\ncopyright (c) \\%s- \\%s VERSANT Corporation -- The copyright and version mes- sage
12451	VBACKUP_CLI__LIST_HEADER	Listing Versant files on device %s: The header printed before starting the list
12452	VBACKUP_CLI__LIST_FORMAT_WITH _COMMENT	Position %d: %s volume %d block- ing %s. %s The format for listing a file which has a comment.
12453	VBACKUP_CLI__LIST_FORMAT_SANS _COMMENT	Position %d: %s volume %d block- ing %s. The format for listing a file which does not have a com- ment.
12454	VBACKUP_CLI__LIST_TRAILER	Done -- The trailer printed after a listing is complete.
12460	VBACKUP_CLI__BACKUP_HEADER	Backing up database %s to device %s -- Message displayed before backup
12461	VBACKUP_CLI__BACKUP_TRAILER	Backup has completed success- fully -- Message displayed after backup.
12462	VBACKUP_CLI__RESTORE_HEADER	Restoring database %s from device %s -- Message displayed before restore
12463	VBACKUP_CLI__RESTORE_TRAILER	Restore has completed success- fully -- Message displayed after backup.
12464	VBACKUP_CLI__PERCENT_GAUGE	0\\%                      50\\% 100\\%\\n                                                  \\n The percentage gauge printed above the percent markers.
12465	VBACKUP_CLI__PERCENT_GAUGE _MARKER	. -- The string used to mark off an increment on the percent gauge.
12466	VBACKUP_CLI__PERCENT_GAUGE _LENGTH	40 -- The number of times the gauge string must be printed to reach 100%

---

---

12467	VBACKUP_CLI__INFO_HEADER	Backup information for database %s -- The message displayed before the backupinfo is listed
12468	VBACKUP_CLI__INFO_FORMAT_WITH _COMMENT	Level %d backup made on %s. %s - - The format for providing info on a database with a comment
12469	VBACKUP_CLI__INFO_FORMAT_SANS _COMMENT	Level %d backup made on %s. -- The format for providing info on a database which does not have a comment.
12470	VBACKUP_CLI__INFO_TRAILER	Done -- The message displayed after the backupinfo is listed
12471	VBACKUP_CLI__PRESS_RETURN	Press <return> when you are ready to continue... -- The mes- sage displayed while vbackup is waiting for a <return>
12472	VBACKUP_CLI__ANOTHER_LEVEL _QUERY	Would you like to do another level of restore\non database %s? [ default = no ] -- asks the user if they would like to do another level of restore
12473	VBACKUP_CLI__MUST_ANSWER_YES _OR_NO	Answer either `yes' or `no' : If the user enters something other than yes or no
12474	VBACKUP_CLI__YES	Yes    %s -- yes
12475	VBACKUP_CLI__NO	No -- no
12476	VBACKUP_CLI__MOUNT_THE_NEXT _TAPE	Mount the appropriate tape and press return to begin... Printed between restores
12477	VBACKUP_CLI__DUBIOUS_DATABASE _NAME	Warning: Dubious database name %s. -- printed for database names beginning with '-'
12478	VBACKUP_CLI__ROLL_FORWARD _QUERY	Would you like to roll forward archived logs for\ndatabase %s? (yes or no) -- asks the user if they would like to do roll for- ward restore

---

12479	VBACKUP_CLI__SETTINGS	Current settings are : \n\n device = %s\n position = %s\n capacity = %s\n block- ing = %s
12480	VBACKUP_CLI__SETTINGS_PROMPT	[?=help]
12481	VBACKUP_CLI__SETTINGS_HELP	Change a setting by typing a name and new value (e.g. "position 2"),\n type "quit" to quit, or type <return> to proceed. It is only necessary\n to type enough of a command to make it unique, so you can type "q"\n instead of "quit" or "p 2" instead of "posi- tion 2".
12482	VBACKUP_CLI__QUIT	Quit -- quit
12483	VBACKUP_CLI__SCRIPT	User input required. Running the following script and exit- ing :\n %s
12484	VBACKUP_CLI__LOG_HEADER	Archiving log records to device %s -- Message displayed before backup
12485	VBACKUP_CLI__RETURN_TO_EXIT	Press <return> when you are ready to exit -- Return to exit
12486	VBACKUP_CLI__SETTINGS_MESSAGE	Change additional settings or type <return> to proceed.
12487	VBACKUP_CLI__RF_ON_INFO	Log archiving is on; but no archiver process is running.\n The next file number will be #d.
12488	VBACKUP_CLI__RF_OFF_INFO	Log archiving currently off.
12489	VBACKUP_CLI__REMOVE_COPY	After you have verified that the database was\n successfully restored; remember to remove\n %s
12490	VBACKUP_CLI__RF_RUNNING_INFO	Currently archiving logs to file #d via connection %d
12491	HABACKUP_CLI__COPYRIGHT	VERSANT Utility HABACKUP Ver- sion %s.%s -\n %s\n copyright (c) %s-%s VERSANT Corporation -- The copyright and version message

---

12500	VBACKUP_RF__CANT_CONNECT	Error while connecting to database %s :\n %s.-- error from o_beginsession or o_connectdb
12501	VBACKUP_RF__CANT_TURN_ON	Error while turning on log roll forward for database %s:\n %s -- error from ut_rollforwardset(db, UT_LRF_ON)
12502	VBACKUP_RF__CANT_TURN_OFF	Error while turning off log roll forward for database %s:\n %s -- error from ut_rollforwardset(db, UT_LRF_OFF)
12503	VBACKUP_RF__CANT_END_SESSION	Error while ending session:\n %s -- error from o_endsession
12504	VBACKUP_RF__WRONG_FILE_NUM	Requested log archive file %#d, received %#d -- error from o_endsession
12505	VBACKUP_RF__WRONG_DBNAME	This is not a log archive for database %d. Error from o_endsession
12506	VBACKUP_RF__PROMPT	Insert log archive %#d of database %s. Prompt for tape
12507	VBACKUP_RF__ABORT	Aborted at user's request.
12508	VBACKUP_RF__LOGGING _TERMINATED	Log archiving terminated for database %d:\n %s
12509	VBACKUP_RF__BACKUP_PENDING	Database %s is pending backup.
12510	VBACKUP_RF__BACKUP_STARTED	Database %s is now being archived.
12511	VBACKUP_RF__COPY_QUERY	During roll forward would you like to apply records\nfrom the database's current log file in addition to any\narchived records ? [default = yes ]
12512	VBACKUP_RF__COPY_QUERY_OLD	This logical log file must then be copied to a safe\nlocation so it is not overwritten during the restore.\n\nenter the path of the logical log [default=%s ]

---

12513	VBACKUP_RF__COPY_QUERY_NEW	Enter the path for storing the copy [ default = %s ]
12514	VBACKUP_RF__COPY_OLD_NOT_FOUND	Could not find the old logical log %s.\n
12515	VBACKUP_RF__COPY_COPY_EXISTS	A file already exists at %s.\n
12516	VBACKUP_RF__COPY_COPYING	Preserving %s as %s.
12517	VBACKUP_RF__VOL_REPLACING	Additional volumes exist in the path.Do u want to overwrite them
12518	VBACKUP_RF__LOGGING_CANT_TERMINATE	Error while turning off log archiving for database %s:\n %s -- error from ut_rollforwardset (db, UT_LRF_NOLOG)
12519	HABACKUP_RF__LLOG_QUERY	Do you have a copy of logical.log file that you\nwould like to apply at this point ? [default = yes ]
12520	HABACKUP_RF__LLOG_PATH	Enter the path of logical.log:
12521	HABACKUP_RF__LLOG_NOT_FOUND	Could not read logical.log at %s.\n
12522	VBACKUP_RF__SUSPEND	Suspended at user's request.
12523	VBACKUP_RF__IS_NOT_SUSPENDED	The database is not in restore suspended mode and therefore denies to resume the data-base.\n In emergency case, please call Versant support immediately.
12550	VBACKUP_SPLIT__CANT_CONNECT	Error while connecting to data-base: %s\n -- error from o_beginsession or o_connectdb
12551	VBACKUP_SPLIT__CANT_DO_CONDEF	Error while connecting to data-base: %s\n -- error from db_namecondef
12552	VBACKUP_SPLIT__CANT_RF_FILENUMBERGET	Error while connecting to data-base: %s\n -- error from ut_rollforwardfilenumget
12553	VBACKUP_SPLIT__CANT_BEGINSPLIT	Error trying to start a split on database: %s\n -- error from net_beginsplit

---

12554	VBACKUP_SPLIT__CANT_ENDSPLIT	Error trying to end a split on database: %s\n -- error from net_endsplit
12555	VBACKUP_SPLIT__CANT_RECOVERSPLIT	Error trying to recover from split failure on database: %s\n -- error from net_recoversplit
12556	VBACKUP_SPLIT__EXEC_FAILURE	Error executing command: %s\n -- error from executecmd
12600	VSTATS_CLI__USAGE	Usage: vstats [options] command\n\n where options are one or more of:\n -filename <filename>\n -stdin\n -database <dbname>\n -username <user>\n -password <passwd>\n -id <connection-id>\n -period <seconds>\n -iterations <iterations>\n -noinfo\n\n and command is one of:\n -stats <stats>\n -on <stats>\n -off <stats>\n -connections\n -transactions\n -locks\n -summary The usage message displayed when arguments aren't entered properly.
12601	VSTATS_CLI__COPYRIGHT	VERSANT Utility VSTATS Version %s.\%s %s\ncopyright (c) %s-%s VERSANT Corporation -- The copyright and version message
12602	VSTATS_CLI__FE_SUMMARY	Frontend per process statistics -- header for fe stats summary
12603	VSTATS_CLI__SE_SUMMARY	Frontend per session statistics -- header for se stats summary
12604	VSTATS_CLI__BE_SUMMARY	Backend per connection statistics -- header for be stats summary
12605	VSTATS_CLI__DB_SUMMARY	Backend per database statistics -- header for db stats summary
12606	VSTATS_CLI__UNIMPLEMENTED	This feature is not implemented yet -- for unimplemented features

---

12607	VSTATS_CLI__STAT_UNDEFINED	The statistic %s is not defined - - for undefined statistics
12608	VSTATS_CLI__NO_DBNAME	You must specify a database name for this command -- given when a dbname is required but not given
12609	VSTATS_CLI__CONNECTIONS _HEADING	Connection ID to database %s:\n -- Heading for list of connec- tions
12610	VSTATS_CLI__CONNECTIONS_ENTRY	Connection ID = %d\n User Name = %s\n Session Name = %s\n Long Transaction = %s\n Server Process = %s:%d\n Client Process = %s:%d -- Format for connections list entry
12611	VSTATS_CLI__CONFIG_OPEN	Could not open configuration file %s -- could not open config file
12612	VSTATS_CLI__CONFIG_ERROR	Error on line %d of %s -- error in config file
12613	VSTATS_CLI__DERIVED_STATS _SUMMARY	Derived statistics from %s: -- header for printing derived stats
12614	VSTATS_CLI__NO_DERIVED_STATS	<none defined> -- for when no derived stats are defined
12615	VSTATS_CLI__COMPOUND_CONFIG _ERROR	Error on line %d of %s:\n\n%s -- error in config file
12616	VSTATS_CLI__STATS_REQUIRED	The '-stats' command requires that you specify at least one statistic -- error in config file
12617	VSTATS_CLI__NO_CONNECTIONS	<no connections> -- printed when there are no connections
12618	VSTATS_CLI__INVALID _CONNECTION_ID	%d is not currently a valid con- nection ID -- printed when there are no connections



---

12619	VSTATS_CLI__CONNECTIONS _ENTRY_TCP	Protocol = TCP/IP\n Server Port = %d.%d.%d.%d:%d\n Client Port = %d.%d.%d.%d:%d -- Format for TCP/IP specific pro- tocol info
12620	VSTATS_CLI__CONNECTIONS_ENTRY _SINGLE	Protocol = Single Process -- Format for 1P specific protocol info
12621	VSTATS_CLI__LOCKS_ENTRY	Object = %s\n External Mode = %s\n Internal Mode = %s\n Transaction ID = %s\n Flags = %s -- Format for locks list entry
12622	VSTATS_CLI__NO_LOCKS	<no locks> -- printed when there are no locks
12623	VSTATS_CLI__LOCK_NOLOCK	None
12624	VSTATS_CLI__LOCK_WLOCK	Write
12625	VSTATS_CLI__LOCK_ULOCK	Update
12626	VSTATS_CLI__LOCK_RLOCK	Read
12627	VSTATS_CLI__LOCK_IWLOCK	Intention to Write
12628	VSTATS_CLI__LOCK_IRLOCK	Intention to Read
12629	VSTATS_CLI__LOCK_RIWLOCK	Read with Intention to Write
12630	VSTATS_CLI__LOCK_RIRLOCK	Read with Intention to Read
12631	VSTATS_CLI__LOCK_RUNNING	Running
12632	VSTATS_CLI__LOCK_WAITING	Waiting
12633	VSTATS_CLI__LOCK_TRANSIENT	Transient
12634	VSTATS_CLI__LOCK_PERSISTENT	Persistent
12635	VSTATS_CLI__LOCKS_HEADING	Locks in database %s:\n -- Head- ing for list of locks
12636	VSTATS_CLI__TRANSACTIONS _ENTRY	Transaction ID = %s\n Name = %s\n Lock Count = %d\n Connection ID = %d\n Flags = %s -- Format for locks list entry
12637	VSTATS_CLI__NO_TRANSACTIONS	<no transactions> -- printed when there are no transactions

---

12638	VSTATS_CLI__TRANSACTIONS _HEADING	Transactions in database %s :\n -- Heading for list of transactions
12639	VSTATS_CLI__TRANSACTIONS _SHORT	Short
12641	VSTATS_CLI__CONNECTIONS_ENTRY _INTERNAL	Protocol = INTERNAL  Format for INTERNAL protocol info
12646	VSTATS_SYMBOL__UNDEFINED	Symbol %s is not defined -- symbol not defined
12647	VSTATS_FUNC__NO_DELTA_ENTRY	No delta entry yet... -- delta entry not found
12648	VSTATS_FUNC__SYMBOL_TYPE _MISMATCH	Argument %s of function %s must be called with a symbol value -- symbol arg type mismatch
12649	VSTATS_FUNC__DOUBLE_TYPE _MISMATCH	Argument %s of function %s must be called with a numeric value -- double arg type mismatch
12650	VSTATS_FUNC__ARG_NOT_FOUND	Attempt to get value of non-existent argument: Arg not found
12651	VSTATS_INPUT__FILE_OPEN	OS error %d while opening file %s -- fopen failed.
12652	VSTATS_INPUT__VERSION	VSTATS %d.%d.%d cannot read %d.%d.%d files --incompatible versions
12653	VSTATS_INPUT__SYNTAX_ERROR	Bad file format -- Bad format
12654	VSTATS_INPUT__EOF	Unexpected end of file while reading %s -- read error
12655	VSTATS_INPUT__ALREADY_OPEN	Attempt to open an open file -- double open()
12656	VSTATS_INPUT__ALREADY_CLOSED	Attempt to close a closed file -- double close()
12657	VSTATS_INPUT__END_OF_INPUT	End of file reached -- expected end of file
12676	VSTATS_PARSER__SYNTAX_ERROR	Syntax Error in position %d of %s -- for syntax errors evaluating expressions

---

12677	VSTATS_PARSER__INTERNAL_ERROR	Internal error while parsing -- This should never happen
12678	VSTATS_PARSER__UNDEFINED _SYMBOL	The symbol %s is not defined -- for undefined symbols
12679	VSTATS_PARSER__BAD _DECLARATION	Error at position %d of declara- tion %s -- for bad declarations
12680	VSTATS_PARSER__NEEDS_DBNAME	In the expression:\n\n %s\n\nthe statistic %s must be followed by a database name. \nall backend and database sta- tistics must do this. (For exam- ple, \n'db_net_bytes_read db1@foo') -- for when db or be stats are missing a dbname
12681	VSTATS_PARSER__NO_BE	In the expression : \n\n %s\n\nyou used a BE statistic (%s) without specifying\na con- nection with the -id option -- for when db or be stats are miss- ing a dbname
12682	VSTATS_PARSER__NEEDS_DBNAME _OR_ESCAPE	In the expression:\n\n %s\n\nthe statistic %s must be followed by a database name. \nall backend and database sta- tistics must do this. (For exam- ple, \n'db_net_bytes_read db1@foo'). If your database name begins with\nanything other than a letter, you must escape it by inserting a\nleading '\\\n' -- for when db or be stats are missing a dbname

## C++/VERSANT ERRORS, SERIES – 8000, 40000

8001	CXX_CANT_CAST	Can not cast object of type %s to type %s. You attempted to cast a pointer or a link to some other type of pointer that it is not compatible with. This often occurs when using the AS() or L_AS() macros. It can also happen if a parameterized collection class contains an object of the wrong class, although C++ compile-time type-safety should prevent this. [ used by ` pclass::offsetof(pclass *type)'] [ used by ` the AS macro, the L_AS macro, pobject::as()']
------	---------------	--

This error can occur if you are trying to cast a pointer to a transient instance of a class and the constructor for the class does not contain the line `VPP_CLASS_CONSTRUCTOR1(ClassName)`, which is needed for runtime type identification.

8002	CXX_NO_CONTAINER	No containers named %s were found on database %s [ used by ` class Container']
8003	CXX_DUP_CONTAINER	Multiple containers named %s were found in database %s [ used by ` class Container']
8004	CXX_BAD_DATE	The date %s %d, %d is illegal [ used by ` class Date (vnih)']
8005	CXX_BAD_DAY	%s is not the name of a day of the week [ used by ` class Date (vnih)']
8006	CXX_NOT_PERSISTENT_VEC	Cannot convert % object to phandle_vec as it contains transient elements
8007	CXX_NOT_VERSION	%s method invoked on non-version %s object
8008	CXX_BAD_ALLOC_SIZE	Allocated %s size must be positive

---

8009	CXX_BAD_INDEX	Index of %d applied to %s object is out of bounds %d [ used by ` Vstr<>' ] [ used by ` varray<>, vlist<> ' ] [ used by ` linkedlist (vnih)']
8010	CXX_NOT_FOUND	%s object was not found in %s for operation %s
8011	CXX_REMOVE_ERR	%s object to be removed was not found in %s [ used by ` class Set (vnih)']
8012	CXX_BAD_PRED	Illegal predicate %s was applied to class %s [ used by ` old pclass::select()']
8013	CXX_BAD_ATTR	%s is not an attribute of class %s [ used by ` old pclass::select()']
8014	CXX_SUBSTRING_ERR	Illegal substring from position %d length %d on %s of length %d [ used by ` class substring (vnih)']
8015	CXX_BAD_CLASS	Got object of class %s when class %s was expected
8016	CXX_BAD_SPECIES	Got object of species %s when species %s was expected
8017	CXX_BAD_ARG_CLASS	Got argument of class %s when class %s was expected by method %s
8018	CXX_BAD_ARG_SPECIES	Got argument of species %s when species %s was expected by method %s
8019	CXX_BAD_RANGE	Range from %d of length %d in %s method %s is invalid [ used by ` class String, substring (vnih)']
8020	CXX_BAD_DATE_RANGE	The date %s %d, %d is outside the legal range [ used by ` class Time (vnih)']

---

8021	CXX_EMPTY	The %s is empty for operation %s [ used by ` class linkedlist (vnih)'] [ used by ` class orderedcltn (vnih)'] [ used by ` class velist<>']
8022	CXX_BAD_TIME	%d,%d,%d %s %s %d,%d is an invalid time [ used by ` class Time (vnih)']
8024	CXX_BAD_MONTH	No such month as %s [ used by ` class Date (vnih)']
8025	CXX_BAD_MONTH_NUM	%d is a bad month number [ used by ` class Date (vnih)']
8026	CXX_BAD_DAY_NUM	%d is a bad day number [ used by `class Date (vnih)']
8027	CXX_DUPLICATE_KEY	Attempt to use duplicate %s key with %s value in %s
8028	CXX_KEY_NOT_FOUND	%s key not found in %s. A key could not be found in a dictio- nary. [ used by ` class Dictio- nary (vnih)'] [ used by ` class V??Dictio- nary<>']
8029	CXX_UNKNOWN	Unknown error number %d was thrown
8030	CXX_ZERO_DENOM	Zero denominator in %s [ used by ` class Fraction (vnih)']
8031	CXX_NO_MEM	Local memory is exhausted
8032	CXX_FUNC_OVER	Double %g conversion to %s yielded large exponent %d [ used by ` class Fraction (vnih)']
8033	CXX_FUNC_UNDEFINED	Double %g conversions is unde- fined for exponent %d [ used by ` class Fraction (vnih)']
8034	CXX_BAD_LINK	%s next should be zero [ used by ` class Link (vnih)']

---

8035	CXX_DERIVED	Virtual method %s should have been implemented by derived class %s. It is the responsibility of subclasses of lookupkey to redefine virtual methods value() and value(const Object&). [ used by `class lookupkey (vnih)']
8036	CXX_SHOULD_NOT	%s should not implement method %s. The operation cannot be done on an instance of this class. [ used by `vnih classes']
8037	CXX_NOT_IMPLEMENTED	Method %s is not implemented in this release
8038	CXX_INVALID_ITERATOR	Iterator is invalid in %s [ used by `VERSANT Collection classes']
8039	CXX_NO_CLSOBJ	Database does not know about class %s. The schema for this class has not been installed into the database. Use sch2db to put the schema from the .sch file into the database.
8040	CXX_NO_CXXCLS	C++ interface does not know about class %s. There is no pclassobj<> for the class compiled into your application, and yet your running application has encountered an instance of that class, or for some other reason needs to find the pclassobj<> for that class.

Your application does not know the definition of a class.

Versant provides class definition information at build-time by embedding within an application the C++ class object for each class used by the application. For example, for class X, the class object PClassObject<X> will be embedded within the executable.

To embed a class object during the compiling and linking process, the application source code must include the header file for the class.

A quick patch for this problem is to put into your main program the following dummy function,

```
dummyFunction() {
```

```
new (PClassObject<X>,,Pointer()) X;
// all classes whose objects may be referenced
```

This dummy function need not be called by the application.

This is a problem because C++ compilers often do not delete temporary objects as soon as they should.

**Please refer to the C++/Versant Reference Manual for further information.**

8041	CXX_CORRUPT_CLSOBJ	Database contains corrupt class %s object
8042	CXX_NO_SUCH_ATTR	No such attribute %s in class %s
8043	CXX_BAD_DOMAIN	Cannot select on domain of attribute %s in class %s
8044	CXX_COMPARE_UNDEFINED	Comparison method undefined. It is the responsibility of a subclass of pvirtual to define a compare method.  This error is thrown by the default pvirtual::compare()
8045	CXX_HASH_UNDEFINED	Hashing method undefined. It is the responsibility of a subclass of pvirtual to define a hash method. This error is thrown by the default pvirtual::hash().
8046	CXX_INTERNAL_ASSERT	C++ interface internal assertion failed. This error should never happen. Report it to Versant.
8047	CXX_THIS_NULL	Got NULL for this in method %s
8048	CXX_BAD_F_ARG	Bad argument for %s in function %s
8049	CXX_BAD_M_ARG	Bad argument for %s in method %s
8050	CXX_ZERO_F_ARG	Bad zero argument for %s in function %s
8051	CXX_ZERO_M_ARG	Bad zero argument for %s in method %s



---

8052	CXX_OM_PANIC	Internal object manager panic: %s (line %d file %s). This is a serious error. No other database object manager operations will be possible for this process after this error. This is cause for terminating your program and trying again. It may indicate database corruption.
8054	CXX_CLASS_SIGNATURE	<p>For class %s, compiled class signature %x does not match schema signature %x in database %s A 32-bit signature is calculated for each class during schema capture using a CRC function across the class and its attributes. This 32-bit signature is compiled into your application and is also deposited in the .sch file.</p> <p>Sch2db will read it from the .sch file, and put it into the database. When your application runs, the signatures in your program are compared against the signatures in the database whenever database classes are used.</p> <p>If the signatures do not match, this exception is throw. This can happen when you have a different defintion of the classes in your application than there are in the database, because you installed the wrong .sch files into the database. To correct the problem, use sch2db to install the correct .sch files into the databases.</p>
8055	CXX_NOT_MUNCHED	Problem in munch -- static objects did not construct. The binary you are executing did not "munch" correctly when it was compiled. It will have to be recompiled properly before it can be used.

8056 CXX\_ASSERTT

An assert macro failed (see "paux.h")The header paux.h contains two macros, assertt() and assertts(), which throw this error when they fail. "assertt()" puts the source filename and linenumber in the error, and assertts() puts a second argument, a string, in the error. Versant uses assertts() inside their class library at positions that should never fail. These will print the extra argument beginning with the word "bomb\_" if they do fail. Contact Versant if one of our assertions fails.

8057 CXX\_OM\_VSTR\_OUTSIDE\_SESSION

A vstr that was created during a session was released or resized after endsession. Links and Vstrs created during a session can only be used during that session. In this case a vstr that was created during a session was changed or released after the session. Often this happens when a transient object (perhaps a spurious temporary object) is created during a session, and then is deleted after endsession.

Often you can avoid this by putting braces { } around the work within a session, and closing them before calling ensession:

```
Dom->beginsession()  
{ ..... Your work ..... }  
Dom->endsession()
```

---

8058	CXX_BOMB	<p>This is a problem because C++ compilers often do not delete. These temporary objects as soon as they could.</p> <p>Internal assertion failed, %s. This error should never occur. Report it to Versant.</p>
------	----------	---

This error can indicate that there is an inconsistency in the build process, in which case you should check your makefiles and make sure that the build is done correctly.

During the template instantiation process for cfront 3.0 based compilers, Versant generates intermediate files of the general form `__PClass_something.cc`. This error can mean that the generation of a file of this form is not controlled by your makefiles, which means that if a class definition changes, these files are not automatically rebuilt. If this is the case, you need to clean out all the intermediate files, the `.o` files, and rebuild from scratch.

8061	CHL_NO_SESSION	<p>A link was used while no session exists. You cannot use links (or phandle or PCOD* or o_object) outside of a session, because the object cache does not exist except during a session. (Likewise, a link from one session is invalid in another session.)</p>
8062	CHL_XCOD	<p>A COD is referenced outside of COD-space. You seem to have a bad link (or phandle or PCOD* or o_object), because it points to an address which cannot possibly be a COD, because it is outside of the range of addresses in which a COD could be formed.</p>
8063	CHL_NO_COD_TYPE	<p>Problem in COD. You seem to have a bad link (or phandle or PCOD* or o_object).</p>
8064	CHL_DELETED	<p>A link to a deleted object was used. This error is not actually enforced; it seems to be too useful to have links to deleted objects. (You are still using a link to an object which has been deleted.)</p>

8065	CHL_TOP_XHEAP	Persistent object located outside object space. You seem to have a bad link (or phandle or PCOD* or o_object). We intuit that because the object associated with this non-link would be located outside of the range of permissible addresses for persistent objects.
8066	CHL_BAD_LOOP	Problem in COD. You seem to have a bad link (or phandle or PCOD* or o_object).
8067	CHH_NULL	Null object header pointer. Something is wrong which lead us to beleive that an object has a null header pointer, which cannot happen. You probably have trashed your object or link.
8068	CHH_NO_CLASS	Problem in object header
8069	CHH_CLASS_XCOD	Problem in object header
8070	CHH2_XHEAP	Problem in extended object header
8071	CHH2_XCOD	Problem in extended object header
8072	CHH_XHEAP	Object header not in heap
8073	CHP_WHAT	Object has bad pobject::what slot
8074	CH_ADDR	Pointer has illegal value
8075	CHV_S_FREED	Vstr has already been freed
8076	CHV_S_TFREED	Transient vstr has already been freed
8077	CHV_S_BOGUS	Vstr pointer does not point to a vstr
8078	CHP_PERSIST	Use of persistent object located outside of heap
8079	CHL_MUST_PERSIST	Link to transient object where persistent object is required
8080	CHV_MUST_PERSIST	Transient vstr where persistent vstr is required
8090	CXX_BAD_VIRTUAL_EVENT	Virtual event %d not understood by object Experimental.
8091	CXX_DOM_IS_NULL	Global variable dom is NULL

---

8092	CXX_REL_MISMATCH	Release %s calls %s of release %s. Two different releases are being mixed together. This will probably cause trouble. The thing that was called is complaining that the caller is of a different release.
8094	CXX_IS_TRANSIENT	Transient object used in persistent object operation. Some operations can only be done on persistent objects. One of these operations was attempted on a transient object. [ used by `vpp scap']
8095	CXX_BILINK_NOTFOUND	A bilink of bilinkvstr relation is missing in the bilink table. Some bilink or bilinkvstr relation is not inserted into the bilink relation table.
8096	CXX_ERROR_HEAD_FMT	<Error %ld the output format of an error number.
8097	CXX_FNAME_LNUM_FMT	In %s line %ld the output format of a file name and line number.
8098	CXX_UNIX_ERRNO_FMT	%s (unix errno %d)the output format of a unix error number and its name
8099	CXX_UNKNOW_ERRNO_FMT	[unknown unix error %d] the output format of an unknown unix error number
8100	CXX_NO_MEMORY	No memory (cannot malloc), no more memory is available
8101	CXX_UNEXPECTED	Unexpected exception, an unexpected exception is caught
8102	CXX_TERMINATING	Terminating, Process is terminating
8103	CXX_TRY_DEPTH	Too many nested try blocks
8104	CXX_DOUBLE_FAULT	Perror thrown while throwing Perror. Perror thrown while throwing perror, an internal error
8115	CXX_WORD_IN	In, the English word "in"

8120	CXX_EMPTY_ARRAY	Viarray(vvarray) is empty at %d for %s please use is_null() to test whether this cell is empty or not before using the operator[]
8121	CXX_EMPTY_LIST	Vilist(vvlist) is empty at %d for %s please use is_null() to test whether this cell is empty or not before using the operator[]
8122	CXX_VDICTION_BASE_CLASS	Vdictionary base class wrong.
8123	CXX_INVALID_METHOD_IN_MULTISESSION	Method is not a valid method in multi-session application
8124	CXX_SELECT_NOT_PERSISTENT	Cannot select from a non-persistent collection object
8125	CXX_TYPE_MISMATCH	Object size of type %s is different from the size in schema object check whether you have rebuilt the schema file after modifying class definition
8126	CXX_PARSE_FAILED	Failed to parse query in d_VQL_Query.
8127	CXX_INVALID_QUERYCOMMAND	D_OQL_execute does not support commands other than O_QRY_SELECT.
8128	CXX_INVALID_PROJECTION	Invalid number of projections in d_OQL_Query::parsedresult.
8129	CXX_ATTRIBUTEOBJ_NOT_FOUND	Attribute object for %s can not be not found.
8130	CXX_ATTRIBUTES_NOT_MATCH	Type of the %d th attribute in projections does not match the type of the %d th element in the result variable.
8131	CXX_OLIST_NOT_SUPPORTED	Type O_LIST is not supported.
8132	CXX_CONVERSION_NOT_ALLOWED	Conversion from %s to %s is not allowed in Pbuffer::convert-type().
8133	CXX_NO_MORE_SUBSTITUTES	No more parameter substitute are allowed.
8134	CXX_SUBSTITUTES_NEEDED	Need parameter substitutes for query.
8135	CXX_PROJECTION_NEEDED	No projection is found in the query.

---

8136	CXX_NOT_MATCH_PROJECTION_TYPE	Result type does not match the type of %d the projection in the query.
8137	CXX_FAIL_TO_MAP_INT_TYPE	Failed to map type INT or UNSIGNED INT to a Versant Type.
8138	CXX_NEED_PARSEQUERY	Need to parse query in d_VQL_Query.
8139	CXX_OQL_CLASS_NOT_FOUND	D_oql_execute() Class Object of Input Type is not found.
8140	CXX_THREAD_NOT_IN_SESSION	There is no session attached to the current vthread object.
8141	CXX_SESSION_NOT_ATTACHED	Session %s is not attached to the current vthread.
8142	CXX_CONTAINER_IS_EMPTY	STL method %s cannot be applied on an empty container.
8143	CXX_ITERATOR_OUT_OF_RANGE	Iterator or pointer points to memory which is out of current allocated space
8144	CXX_ITERATOR_NOT_IN_ORDER	The iterator pair(first last) has the wrong order.
8145	CXX_NO_VTHREAD_OBJECT	There is no vthread object attached to current session
8146	CXX_SESSION_ATTACHED_ALREADY	Session init fails because current thread has already attached to another session
8147	CXX_TLS_KEY_INIT_ERROR	Kernel call vs_tlsgetptr() returns NULL. Cannot initialize thread specific data key
8148	CXX_TLS_ALLOCATION_ERROR	Cannot allocate thread local storage. Vs_tlsgetptr() failed.
8149	CXX_KERNEL_CLSOBJ_NOT_INITIALIZED	Cannot access kernel class object; session may not start yet
8150	CXX_PUSH_EMPTY_ITEM_TO_LIST	Try to push an empty item to a list
8151	CXX_TOO_MANY_SESSIONS	Try to open more than maximum number of sessions
8152	CXX_LOCAL_TIME_EXCEEDS_LIMITATION	Cannot convert local time earlier than Jan 1st 1970
8153	CXX_NULL_PROFILE_POINTER	Cannot accept NULL pointer in readprofile

---

8154	CXX_QUERY_HANDLE_NULL	Query handle is NULL. Query may not be compiled
8155	CXX_RESULT_HANDLE_NULL	Result handle is NULL
8156	CXX_NOT_ALL_PROJECTION_SUPPORTED	Only O_OBJECT_TYPE projections are supported
8502	SCAP_WORD_FATAL	Error
8504	SCAP_WORD_WARN	Warning
8505	SCAP_WORD_NOTE	Note
8511	SCAP_CANNOT_CREAT_EMPTY	Cannot create file with EMPTY name
8512	SCAP_CANNOT_CREAT	Cannot create file %s
8513	SCAP_CANNOT_OPEN	Cannot open file %s
8514	SCAP_CANNOT_OPEN_EMPTY	Cannot open file with EMPTY name
8518	SCAP_NEED_DB_ARG	You must specify a database name after the %d option.

A command-line argument naming a database was missing for a Schema Utility.

8519	SCAP_MALLOC	Out of memory, cannot malloc. The process ran out of memory while trying to malloc(). You need more swap space. You might terminate some processes to free up some swap space. Also check that you do not have "limit"s set low.
8520	SCAP_BOMB	Internal assertion failed, %s. This error should never occur. Report it to Versant.



8522 SCAP\_VERSION

Incompatible .sch-version number: program=%d/%s .sch=%d/%s. The sch-version number or the machine name in the .sch files is not compatible with this version of sch2db or sch2z. Recompile (or run schema capture again) to generate new .sch files.

You can look at the top line of a .sch file to see the sch-version number and the architecture name. It follows the letter "v": v 113 sparcs. The -v option to sch2db will emit a timestamp ending in the sch-version number, following "/s": .../s113sparcs

The version number built into a schema file is not compatible with the version number of the sch2db or sch2z utility that you are using. The solution is to recompile or rerun schema capture to generate new .sch schema files.

You can look at the top line of a .sch schema file to see the schema version number and the architecture name. The information you want follows the letter "v". For example, v 113 sparcs. Then, to determine the version number of your utility, run it with the -v option, which will emit a timestamp ending in /s plus its schema version number. For example, .../s113sparcs.

**For more information, on Schema Capture utility, please refer to the utility “sch2db” in the chapter "Utilities" in the *C++/Versant Reference Manual*.**

8523 SCAP\_CORRUPT\_SCH

Corrupt .sch file, unexpected object type %d. The input to a schema utility was not a correct .sch file. Perhaps you gave the utility the wrong filename, or perhaps your .sch files have been corrupted.

8524 SCAP\_SEL\_CLS\_CLS

Cannot find classes in database %s. The utility was unable to do a select on instances of class "class". This might be due to locking or deadlock, but it could indicate a corrupt database

8525	SCAP_SETDOMAIN	Unable to fixup forward reference domain, %s class, %s attribute, %s. The domain of some attribute was not defined at the time that the class containing it was defined. At the end of sch2db we try to fix up all of these attributes, but the domain still is not defined, or for some other reason the object manager cannot fix up this attribute.
8526	SCAP_FIXUP	Problem during forward reference fixup -- some classes may not be fixed up. Could be a sign of database corruption. See SCAP_SETDOMAIN.
8527	SCAP_FIXUP_DOMAIN	Cannot locate domain class %s object during fixup of class %s, attribute %s. A class was defined with forward references to the domain class. On the second pass, when we wanted to fixup the forward reference, the domain class still was not defined. You may be missing some .sch files, or your database could be corrupt. See SCAP_SETDOMAIN.
8528	SCAP_VSTR	Vstr operation failed within schema utility. Check the kernel error code. You may need a bigger front end cache size.
8529	SCAP_PREP_CLS	Cannot add signature to class object, for class %s. We were unable to mark the class object as dirty, so we could not add the class signature to it. This will prevent the schema signature assertion at runtime.
8530	SCAP_LOC_CLSOBJ	Cannot locate class object, class %s. This probably indicates a corrupt database, but maybe you were just locked out.

---

8531	SCAP_LOC_ATTR	Cannot locate attribute object, class %s attribute %s. This probably indicates a corrupt database, but maybe you were just locked out.
8532	SCAP_LOC_DOMAIN	Cannot locate domain object, class %s attribute %s. This probably indicates a corrupt database, but maybe you were just locked out.
8533	SCAP_ATTR_MATCH	Database attribute names do not match .sch file, database %s class %s database-attribute %s .sch-attribute %s. This probably indicates a corrupt database.
8534	SCAP_ATTR_OFFSET	Database attribute offsets do not match .sch file, database %s class %s attribute %s database-offset %d .sch-offset %d. This represents an inconsistency between the way cfront aligns attributes and they way the database thinks they should be aligned. Let Versant know about this error.
8535	SCAP_LOC_SUPERCLS	Cannot locate the class object for the superclass of %s. The class is probably not usable in the database until the classes are redefined.
8536	SCAP_DEFCLASS	Could not define class %d
8537	SCAP_DEFCLASS_LOC	Could not locate class object after defining class %s
8538	SCAP_LOC_ATTR_CLASSID	Cannot locate o_attrobj::classid of class %s attr %s
8539	SCAP_LOC_ATTR_ATTRID	Cannot locate o_attrobj::attrid of class %s attr %s
8540	SCAP_CORRUPT_LOCATE	Database corruption detected, cannot use class %s (internal id %s)
8541	SCAP_DROPCLASS	Cannot drop class %s from the database

8542	SCAP_DROPINSTS	Cannot drop instances of class %s from the database
8543	SCAP_DROPATTR	Cannot drop attribute %s from class %s
8544	SCAP_NEWATTRAT	Cannot add new attribute %s to class %d
8545	SCAP_ADD_CLS_SIG	Cannot attach signature %x to database class object %x. For some reason we were unable to locate the class object or to change it. Therefore we could not add the signature to this object. This may be a problem if it already had a signature and is now wrong. If you can figure out why the class was unavailable, and fix it, you should try the schema operation again.
8546	SCAP_DIRTY	Cannot mark an object dirty. A call to the object manager o_preptochange() was unsuccessful. Perhaps we were unable to get a write lock on the object, or were connected to a read-only database.
8547	SCAP_NO_OP_STAR	Operator * undefined in Link-like class %s. If you derive classes from plink, you must provide a monadic "operator *()" (the indirection operator). Schema capture uses this operator to determine what kind of a link it is.
8548	SCAP_NO_OP_BRACKET	Operator * undefined in Link-like class %s. If you derive classes from pvstr, you must provide an "operator *[]" (the subscript operator). Schema capture uses this operator to determine what kind of a vstr it is.

---

8549	SCAP_BITFIELD	Sorry, bitfields not supported in database class %s. Classes derived from pobject cannot contain bitfields. Try defining them as "o_ulb" or "o_u2b" or "o_u4b", as appropriate.
8550	SCAP_VSTR_NOTSIMPLE	Vstr of nonsimple type %s used in definition of class %s. If classes derive from pobject and are going to have persistent instances, they cannot have Vstr's of non-simple data types. "sch2db" will complain if you try to install this class in a database.
8551	SCAP_BEGINSESSION	Cannot begin database session on database %s. The begin session method failed for the given reason. You might try other applications to see if they can access the database. You might have misspelled the database name, or you might not have permission to use it.
8552	SCAP_ENDSESSION	Error during endsession on database %s. An error was returned during the endsession methods on the database. If you made changes to the database and have not enabled logging, you may have lost data.
8553	SCAP_NO_DB_NAME	No database name was provided -- use "-D" option. You need to name which database the command is to act upon. You may do this with the "-D" option, or you can set the O_DBNAME environment variable to the name of the database.

8554	SCAP_SCH_CONFLICT	Two declarations of class %s conflict with each other. Within a group of .sch files, there are two different definitions of the given class. This error can happen if two different modules use two different classes of the same name, or if you recompile one module after making changes, but do not recompile the other module.
8555	SCAP_CLASS_DIFFERS	<p>Class %s already exists in the database in a different form. The named class already exists in the database when a .sch file contained the class in a different form.</p> <p>If you did not specify "-e", you may try again with "-e" if you want to evolve the classes (and instances), or you may use "-f" if you want to drop the old classes (and instances) and define them again.</p> <p>If you specified "-e", no supported methods of evolution could be used to evolve the class. You will have to either make the new class definitions differ in ways we support evolution, or use the "-f" option to drop the old classes (and instances) and define them again.</p>
8556	SCAP_CMDNAME	This Versant command does not understand its name. Some versant commands use thier name to determine what to do. Perhaps you renamed the binary, or have a link to it by a different name than they one used by Versant. The name might ought to be "sch2db" or "dropclass" or "sch2z".

---

8557	SCAP_OPTION	Unknown dash option
8558	SCAP_OBJECT_FILE_MAGIC	Object file format error, wrong magic number. The schema capture utility can only be invoked on IBM RS6000 platform running AIX 3.2
8559	SCAP_OBJECT_COMPILE_DEBUG	Cannot read the .debug section, did you forget to compile with '-schema' option? Did you compile the file with '-g' option? Versant schema capture utility does the schema capture based on the debugging information embedded in the object code. To enable the schema capture, the object code needs to be compiled with '-g' option.
8560	SCAP_DEBUG_MAP	Cannot map the .debug section to memory, please report to Versant
8561	SCAP_SYMBOL_MAP	Cannot map the symbol table section to memory, please report to Versant
8562	SCAP_STAB_FORMAT	Stab format error, please report to Versant
8563	SCAP_OBJECT_FILE_OPEN	Cannot open the object file %s
8564	SCAP_STAB_ALIGNMENT_ERROR	Stab format error, alignment error
8565	SCAP_CANT_OPEN_FILE	Cannot open file %s for write
8566	SCAP_ABORT_IN_S	ABORT in %s ,
8567	SCAP_S_INTERRUPTED	%s Interrupted
8568	SCAP_NO_PLUS_F	Missing +fsourcefilename option
8569	SCAP_ERROR_IN_S	Error in %s
8570	SCAP_NO_PLUS_T	Plus T option not required
8571	SCAP_NO_CXXSUN_SUPPORT	Not supported by cxxsun
8572	SCAP_CXXSUN_NO_SINGLE_QUOTE	Cxxsun does not support single quote in arguments
8573	SCAP_MISSING_IN_PATHFILE	%s path for %s missing from file %s
8574	SCAP_XLC_INTERNAL_ERROR	Internal error, cannot recover, please report to Versant

8575	SCAP_XLC_HEAP_STORAGE	Fatal error in ,,operator new(), heap storage exhausted
8576	SCAP_XLC_SCAP_USAGE	Usage:scapxlc [+v] [+z] [+C] +filename
8577	SCAP_XLC_INPUT_FILE_MISSING	Input file name missing, Usage:scapxlc [+v] [+z] [+C] +filename
8578	SCAP_XLC_HEAP_STORAGE_C	Fatal error in Classdef::operator new(), heap storage exhausted
8579	SCAP_XLC_HEAP_STORAGE_S	Fatal error in Sym::operator new(), heap storage exhausted
8580	SCAP_POINTER_TO_CLASS_MEMBER	Warning, pointer to class data member inside persistent class
8581	SCAP_NO_SCHEMA_FILE_CREATED	No .sch file emitted (missing or bad +f option)
8582	SCAP_CANT_DETERMINE_VTBL	Cannot determine the vtbl name for class %s
8583	SCAP_FATAL_ERROR_IN_BSLOT	Versant Internal error, please report to Versant
8584	SCAP_NO_USER_NAME	No user name was provided -- use "-U" option
8585	SCAP_NO_PASSWORD	No password was provided -- use "-P" option
8601	PERR_CANNOT_WRITE	Cannot write to file, An attempt to write a file failed. Checking the operating system error code (errno) may be helpful
8602	PERR_CANNOT_MAKE_TOOLCODE	Cannot convert error to toolcode string. A toolcode string is an internal format for Versant tools to use. Some error happened while trying to make this string.



---

8701	EVPP_BOMB	<p>Internal error in vpp. An internal error that should never happen has occurred. Please contact Versant. It would help if you compile with the +D option, and send us the "_cpp" file, so we can reproduce the problem. Also write down the "bomb_" code that gets printed with the error.</p> <p>[ used by ` vpp']</p>
8703	EVPP_EXPSUP_NOTCLASS	<p>Cannot VPP_EXPAND_SUPER on non-class. The statement VPP_EXPAND_SUPER is special, and its argument must be a class name. You may be using a non-class name as if it were a class, or vpp could be confused and not realize that something is a class. Customers should not be using the VPP_EXPAND_SUPER statement directly.</p> <p>[ used by ` vpp']</p>
8704	EVPP_EXPSUP_OUTSIDE	<p>Cannot use VPP_EXPAND_SUPER outside Template. The special VPP_EXPAND_SUPER statement can only be used inside a Template definition. Customers should not be using the VPP_EXPAND_SUPER statement directly.</p> <p>[ used by ` vpp']</p>
8705	EVPP_PCO_UNDEF	<p>Template pclassobj undefined. The template pclassobj is not defined. Perhaps you forgot to include the standard headers, or perhaps they are corrupt.</p> <p>[ used by ` vpp']</p>

8706	EVPP_TPL_UNDEF	Unknown Template. The template has not been defined yet. The ability to use forward reference templates is very limited; all templates should be fully defined before they are used. [ used by ` vpp']
8707	EVPP_AFTER_NOT_TPL	After clause is not a Template. Only template occurrences may follow the keyword 'After' in a clause in a template definition. For example, you could say After Link<X> but you could not say After Employee [ used by ` vpp']
8708	EVPP_NEST_TPL	Cannot nest Templates definition. Template definitions may not occur inside a template definition. [ used by ` vpp']
8709	EVPP_TPL_REDEF	Template redefined. A template can only be defined once. [ used by ` vpp']
8710	EVPP_OPTION	Undefined option flag. A command line option to vpp (starting with a dash) was not understood. [ used by ` vpp']
8711	EVPP_ARGV	No arguments needed. The vpp command line does not take any arguments except for those beginning with a dash '-'. [ used by ` vpp']
8712	EVPP_STDOUT	Error writing standard output. The vpp command sends its output to standard output. A write failed. This could be because the disk filled, or due to a hardware error. [ used by ` vpp']

---

8713	EVPP_NO_IMP	There is no Implementation for this Template. You have said to Implement a template that does not have an Implement clause. [ used by ` vpp']
8714	EVPP_CLAUSE	Undefined Template clause. A clause of a Template is undefined. This could be a clause named in another After clause, or it could be a clause that is special to the Versant C++ interface. The latter should not happen if you include pobject.h. [ used by ` vpp']
8715	EVPP_NEST_EXPSUP	Cannot nest VPP_EXPAND_SUPER. Customers should not be using the special VPP_EXPAND_SUPER statement. Versant's own code should never have nested VPP_EXPAND_SUPER statements. So this error should not happen. [ used by ` vpp']
8716	EVPP_OUT_OF_MEMORY	Out of memory in parser. The vpp program ran out of memory. Adding more swap space could help, as could removing some processes from the machine. Using "stopdb" on a Versant database can also help a lot. [ used by ` vpp']
8717	EVPP_SYNTAX_ERROR	General syntax error. The lexer or parser of vpp has detected a syntax error. These are very often mismatched parentheses or curly braces. Some nonstandard C++ constructs can also confuse vpp. [ used by ` vpp parser']

8718	EVPP_MULTIPLE_IMPLEMENT	<p>Multiple implement clauses in template definition. There can only be one Implement clause in a template definition.</p> <p>[ used by ` vpp']</p>
8719	EVPP_PRIMITIVE_UNDEFINED	<p>Primitive template undefined. Some templates are special to the Versant interface, and must be there if you are using the primitive Versant classes. These templates should all be defined if you included pobject.h.</p> <p>[ used by ` vpp']</p>
8720	EVPP_BAD_LINE_MARKER	<p>Bad line marker. The vpp program reads and writes filename-and-linenummer marker lines in the format that the standard C pre-processor understands.</p> <p>Examples: # "filename" 88 or #line "filename" 88</p> <p>Something must have created a marker line that vpp cannot understand.</p> <p>[ used by ` vpp']</p>
8721	EVPP_BAD_PARAMETER	<p>Parameter is not a word. Vpp template occurrences only accept simple C++ identifiers and other template occurrences as parameters.</p> <p>[ used by ` vpp']</p>
8722	EVPP_TPL_RECURSION	<p>Recursive template name. Vpp has seen a template (or a group of templates) that invoke themselves to get longer and longer template names. This can happen if you have recursive templates. For instance, template Foo&lt;bar&gt; which uses Foo&lt;Foo&lt;bar&gt;&gt; which uses Foo&lt;Foo&lt;Foo&lt;bar&gt;&gt;&gt; etc. Would create this error.</p> <p>[ used by ` vpp']</p>

---

8723	EVPP_INFINITE_TDEF	Infinite typedef recursion. The vpp program attempts to understand simple typedef statements so that it will not generate redundant template definitions. This error occurs if the typedefs create an infinite loop. [ used by ` vpp']
8750	EVPP_TOO_MANY_NESTED_CLASSES	VPP abort --- more than %d nested class levels!The vpp parser has a limit of the nesting class level. It is unusual to reach this limit in a normal program. [ used by ` vpp']
8751	EVPP_NO_MEMORY_SCANNER	VPP abort --- out of memory. No memory is available. Either the input file is too huge or there is not enough virtual memory. [used by ` vpp']
8753	EVPP_SYN_ERROR	VPP finds syntax errors in the following files, VPP finds some syntax errors. If these errors are not reported by your C++ compiler, there might be a bug in VPP. [ used by ` vpp']
8754	EVPP_ACTION_UNKNOWN	VPP abort --- internal error, unknow action. This is an internal error. Please report it to Versant. [ used by ` vpp']
8755	EVPP_TOO_MANY_TMPL_ARGUMENT	VPP abort ---- too many template arguments. This is an internal ARGUMENT error. Please report it to Versant. [ used by ` vpp']
8801	CIH_OFFSET_NO_MEMORY_ATTR	Offset calculation function is not able to allocate enough memory for offset table
8802	CIH_INTERNAL_NAME_COMPOSTION_ERROR	Offset calculation function is not able to compose the attribute name, please report to Versant

8803	CIH_OFFSET_FUNCTION_NULL	The pointer for offset calculation in the pclass object is NULL
8804	CIH_OFFSET_NO_MEMORY_OBJECT	Offset calculation function is not able to allocate enough memory for the object
8805	CIH_NO_CXX_ACTIVATOR	The pointer for class activator is NULL. The virtual table pointer and virtual base pointer value cannot be set
8806	CIH_OFFSET_CALCULATION_ERROR	The offset calculation function internal error, cannot allocate enough memory
8807	CXX_TRANSIENT_NO_WHAT	The transient object doesn't have RTTI information. You need to call O_ACTIVATE_RTTI on the transient object.
8808	CXX_ATTRIBUTE_NOT_FOUND	Versant internal error. Attribute %s can not be found in the pclassobject of %s
8900	VRT_BUFFER_SIZE_MISMATCH	Can not interpret an object of size %d as type %s. The vrtbuffer class is used to access heterogeneous attribute values. There is a mismatch between the size of the attribute data type and the size of the data type that application interprets that attribute to be.
8925	VSO_MAGIC_NUMBER_MISMATCH	The shared object arena is corrupted. A magic number is stored in the shared object arena. When an application connects to shared object arena the magic number is checked. The exception is thrown if the magic number is incorrect.
8926	VSO_VERSION_MISMATCH	The format of the shared object arena does not match the application. A format version stored in the shared object arena does not match the number compiled into the application.

---

8927	VSO_CLASS_SIGNATURE_MISMATCH	For class %s, compiled class signature does not match schema signature in the shared object arena A 32-bit signature is calculated for each class during schema capture using a CRC function across the class and its attributes. This 32-bit signature is compiled into your application. The exception is thrown if the class signature for the same class is not the same in all applications which share the same object arena.
40101	T_SCAP_DEFINE_CLASS	Define class %s
40102	T_SCAP_DROP_CLASS	Drop class %s and all its instances
40103	T_SCAP_DROP_INSTANCES	Drop instances of class %s
40104	T_SCAP_GET_NEW_ATTR	Class %s gets new attribute %s
40105	T_SCAP_GET_DROP_ATTR	Class %s drop attribute %s
40106	T_SCAP_PRINT_VERSION	This is version %s
40107	T_SCAP_SCHEMA_CHANGES_NEEDED	Schema changes needed:
40108	T_SCAP_NO_CHANGES_APPLIED	NO CHANGES APPLIED
40109	T_SCAP_APPLYING_SCHEMA_CHANGES	APPLYING SCHEMA CHANGES
40110	T_SCAP_WILL_DROP_INST	Will drop instances of,
40111	T_SCAP_OK_TO_APPLY_Q	Ok to apply changes ?
40112	T_SCAP_NOT_A_TTY	Standard input is not a tty, so '-y' option is required for changes
40113	T_SCAP_DROPPING_INSTANCES	DROPPING INSTANCES
40114	T_SCAP_NO_SCHEMA_CH_REQ	No schema changes required.
40115	T_SCAP_NO_CLASSES_TO_DROP_INST	There are no classes of which to drop the instances
40116	T_SCAP_GET_RENAME_ATTR	Class %s renames attribute %s as %s
40117	T_SCAP_GROUP_ADD_ATTR	Group add %d attributes
40118	T_SCAP_CLASS_NEEDS_REARRANGE	Class %s needs rearrange
40150	T_SCAP_USAGE_DROPCLASS_1	Usage, dropclass [-options....] Class1 class2 ...
40151	T_SCAP_USAGE_DROPINST_1	Usage, dropinst [-options....] Class1 class2 ...

---

40152	T_SCAP_USAGE_SCH2DB_1	Usage, sch2db [-options....] File1.sch file2.sch ...
40153	T_SCAP_USAGE_DASH_D	\t-D db\tddatabase, use database db (or use O_DBNAME environment)
40154	T_SCAP_USAGE_DASH_N	\t-n\tno, make no changes
40155	T_SCAP_USAGE_DASH_Y	\t-y\tyes, make any changes
40156	T_SCAP_USAGE_DASH_I	\t-i\tinteractive (default), ask yes or no (stdin must be tty)
40157	T_SCAP_USAGE_DASH_E	\t-e\tevolution, schema evolu- tion is a potential strategy
40158	T_SCAP_USAGE_DASH_R	\t-r\trename, rename attribute is a potential strategy
40159	T_SCAP_USAGE_DASH_F	\t-f\tforce, dropping classes is a potential strategy
40160	T_SCAP_UNKNOWN_OPTION	Unknown Option
40161	T_SCAP_USAGE_DASH_U	\t-u\tuser, run as this DB user
40162	T_SCAP_USAGE_DASH_P	\t-p\tpassword, enter password for the DB user entered in -u
40171	T_SCAP_DB2TTY_USAGE	\n\tusage 1: db2tty -D data- basename [options] [ classnames ... ]\n\tusage 2: db2tty -D databasename [-l] -o loids\n\t- i\tshow instances\n\t-a\tshow all classes (including system)\n\t- s\tshow schema infomation only\n\t-t LX\toperate in long transaction LX\n\t-l\tuse read locks (default is NOLOCK)\n\t-n <num>\tnumber of objects per cur- sor fetch (default is 200,valid only with -i option)\n\t-o loads\tdisplay objects with spec- ified loads



---

40301	REBILINK_WARNING	\ndo you really want to change all\nold bilink objects in the database?\n\nthe new bilink objects are NOT compatible with old\napplications. You will need to recompile your old\nap- plications to access new bilink objects.\n\npress the ENTER key twice to continue, or press\ncon- trol-C to abort this program.\n
40302	REBILINK_USAGE	%s [-Q] -D database_name\n
40303	REBILINK_LOCATE_SUPER	Problem locating superclass object [%ld] of class %s,
40304	REBILINK_LOCATE_ATTR	Problem locating attribute object [%ld] of class %s,
40401	D_Error_None	
40402	D_Error_databaseclassmismatch	For class %s, compiled class sig- nature %x does not match schema signature %x in database %s
40404	D_Error_databaseclosed	Database %s is either already closed or has not yet been opened
40405	D_Error_databaseopen	Database %s is already opened
40406	D_Error_dateinvalid	%d/%d/%d is an invalid date
40407	D_Error_iteratorexhausted	Iterator is invalid in %s
40408	D_Error_namenotunique	An object with name %s already exists in the database
40409	D_Error_positionoutofrange	Index of %d applied to %s object is out of bounds %d
40410	D_Error_queryparameter	Countinvalid
40411	D_Error_queryparameter	Typeinvalid
40412	D_Error_refinvalid	
40413	D_Error_refnull	D_Ref is NULL
40414	D_Error_timeinvalid	%d,%d,%d is an invalid time
40415	D_Error_timestampinvalid	Invalid timestamp
40416	D_Error_transactionopen	A transaction has already opened
40418	D_Error_databasenull	The database pointer is NULL
40419	D_Error_Iteratoruninitialized	The iterator has not been ini- tialized with a collection
40420	D_Error_tzinvalid	%d,%d is an invalid time zone time

40421	D_Error_Double_Fault	D_Error thrown while throwing d_Error
40422	D_Error_badinterval	D_Error thrown while throwing d_Error, an internal error The d_Interval spcified is bad, check the hour, min and sec
40423	D_Error_Already_Connected	This group database has been connected already
40424	D_Error_sessiondatabaseopened	A session database has already been opened
40425	D_Error_nametransientobject	Cannot create name for a transient object
40426	D_Error_Duplicaterelsetinsertion	Try to insert an duplicated element to a d_Rel_Set
40427	D_Error_elementnotincollection	Cannot find the element in a collection
40428	D_Error_invalideventtype	Invalid event type for function get_notified
40429	D_Error_illegaldatabase name	A null database name is not allowed
40430	D_Error_transactionstillactive	Cannot close a database with active transactions
40431	D_Error _transactionalreadydetached	The transaction had already detached from the database
40432	D_Error _Transactionalreadyattached	The transaction had already attached to the database
40433	D_Error_transactionclosed	A transaction is either closed or not yet opened
40434	D_Error_Nottransactiondatabase	There is no database opened for current transaction
40435	D_Error _transactionnotconnected	Current transaction is not connected to database %s
40436	D_Error_databaseinvalid	Cannot open database %s. Check if the dbname is valid.
40437	D_Error _deleteunknowntransientobj	Cannot delete a transient object of unknown type
40600	D_Error_XA_OK	
40602	D_Error_XAER_ASYNC	Async. Operation already outstanding

---

40603	D_Error_XAER_RMERR	An RM error occurred in the transaction branch
40604	D_Error_XAER_NOTA	The XID is not valid
40605	D_Error_XAER_INVALID	Invalid arguments were given
40606	D_Error_XAER_PROTO	XA routine invoked in improper context
40607	D_Error_XAER_RMFAIL	The Resource Manager is unavailable
40608	D_Error_XAER_DUPID	XID already exists
40609	D_Error_XAER_OUTSIDE	RM doing work outside global transaction
40610	CXX_NO_SESSIONPOOL	The SessionPool is unavailable
50000	USER_DEFINED_ERROR	User defined error is \$(userdefined)

## GUI TOOL ERRORS, SERIES – 10000

10000	GUI__TEST_ERROR_MESSAGES	GUI__TEST_ERROR_MESSAGES. During initialization, this test message is read in. If it does not match the string "GUI__TEST_ERROR_MESSAGES", the program is aborted with a message that the error message file could not be found.
10026	GUI_BOOLEAN__CLASSNAME	Boolean -- The name of the gui_boolean type
10027	GUI_BOOLEAN__TURN_ON	Turn On -- The name of the guiboolean_turnon operation.
10028	GUI_BOOLEAN__TURN_OFF	Turn Off -- The name of the guiboolean_turnoff operation.
10029	GUI_BOOLEAN__TOGGLE	Toggle -- The name of the guiboolean_toggle operation.
10051	GUI_BUTTON__CLASSNAME	Button -- The name of the gui_button type
10061	GUI_CALLBACK__CLASSNAME	Callback -- The name of the gui_callback type
10076	GUI_DIALOG__CLASSNAME	Dialog -- The name of the gui_dialog type
10101	GUI_DICTIONARY__CLASSNAME	Dictionary -- The name of the gui_dictionary type
10126	GUI_EDIT_FIELD__CLASSNAME	Edit Field --The name of the gui_editfield type
10151	GUI_ELEMENT__CLASSNAME	Element -- The name of the gui_element type
10176	GUI_ERROR__CLASSNAME	Error -- The name of the gui_error type
10177	GUI_ERROR__WARNING	Warning -- Format for warnings printed to stdout
10178	GUI_ERROR__ERROR	Error -- Format for errors printed to stdout
10179	GUI_ERROR__FATAL_ERROR	Fatal Error -- Format for fatal errors printed to stdout

---

10201	GUI_FIELD__CLASSNAME	Field -- The name of the gui_field type
10202	GUI_FIELD__CUT	Cut -- The name of the guifield_cut operation
10203	GUI_FIELD__PASTE	Paste -- The name of the guifield_paste operation
10204	GUI_FIELD__CLEAR	Clear -- The name of the guifield_clear operation
10226	GUI_FILE__CLASSNAME	File -- The name of the gui_file type
10251	GUI_FONT__CLASSNAME	Font -- The name of the gui_font type
10252	GUI_FONT__HELVETICA	Helvetica -- The name of the Helvetica font
10253	GUI_FONT__TIMES	Times -- The name of the Times font
10254	GUI_FONT__COURIER	Courier -- The name of the Courier font
10255	GUI_FONT__SYMBOL	Symbol -- The name of the Symbol font
10256	GUI_FONT__BOLD	Bold -- The name of the Bold style
10257	GUI_FONT__ITALIC	Italic -- The name of the Italic style
10258	GUI_FONT__UNDERLINE	Underline -- The name of the Underline style
10276	GUI_IPC__CLASSNAME	GUI IPC -- The name of the gui_ipc type
10277	GUI_IPC__CLEANUP	Cleaning up... -- The message shown during cleanup which happens after ctrl-C
10301	GUI_LABEL__CLASSNAME	Label -- The name of the gui_label type
10302	GUI_LABEL__COPY	Copy -- The name of the guilabel_copy operation
10326	GUI_LIST__CLASSNAME	List -- The name of the gui_button type

10351	GUI_LIST_VIEW__CLASSNAME	List View -- The name of the gui_listview type
10371	GUI_MESSAGING__LOCALIZED_WB_FLAG	-workbench -- A localized alias for -workbench / -softbench
10372	GUI_MESSAGING__WB_NOT_AVAILABLE	This application was not compiled to support Workbench/Softbench. Error given when -workbench is used but the application was not Compiled with workbench support.
10376	GUI_MESSAGE_DIALOG__CLASSNAME	Message Dialog -- The name of the gui_messagedialog type
10401	GUI_MENU__CLASSNAME	Menu -- The name of the gui_menu type
10402	GUI_MENU__HELP_MENU_NAME	Help -- The name of the help menu
10426	GUI_NUMBER__CLASSNAME	Number -- The name of the gui_number type
10427	GUI_NUMBER__SECOND	Second -- The word to use for 1 second
10428	GUI_NUMBER__SECONDS	Seconds -- The word to use for n seconds
10429	GUI_NUMBER__MINUTE	Minute -- The word to use for 1 minute
10430	GUI_NUMBER__MINUTES	Minutes -- The word to use for n minutes
10431	GUI_NUMBER__HOUR	Hour -- The word to use for 1 hour
10432	GUI_NUMBER__HOURS	Hours -- The word to use for n hours
10433	GUI_NUMBER__DAY	Day -- The word to use for 1 day
10434	GUI_NUMBER__DAYS	Days -- The word to use for n days
10435	GUI_NUMBER__BYTE	Byte -- The word to use for 1 byte
10436	GUI_NUMBER__BYTES	Bytes -- The word to use for n bytes

---

10437	GUI_NUMBER__KILOBYTE	Kilobyte -- The word to use for 1 kilobytes
10438	GUI_NUMBER__KILOBYTES	Kilobytes -- The word to use for n kilobytes
10439	GUI_NUMBER__MEGABYTE	Megabyte -- The word to use for 1 megabytes
10440	GUI_NUMBER__MEGABYTES	Megabytes -- The word to use for n megabytes
10441	GUI_NUMBER__GIGABYTE	Gigabyte -- The word to use for 1 gigabytes
10442	GUI_NUMBER__GIGABYTES	Gigabytes -- The word to use for n gigabytes
10451	GUI_OBJECT__CLASSNAME	Object -- The name of the gui_object type
10476	GUI_OPERATION_DIALOG__CLASSNAME	Operation Dialog -- The name of the gui_operationdialog type
10477	GUI_OPERATION_DIALOG__OK	OK -- The name of the guioperationdialog_ok operation used to implement the "OK" button in the operation dialog box.
10478	GUI_OPERATION_DIALOG__OK__EMPTY_FIELD	All the fields must be filled in. -- The message presented to the user when a field is left empty
10479	GUI_OPERATION_DIALOG__CANCEL	Cancel The name of the guioperationdialog_cancel operation used to implement the "Cancel" button in the operation dialog box.
10480	GUI_OPERATION_DIALOG__WARNING	Warning -- The name of the guioperationdialog_warning operation used to implement the warning dialog box for operations with warnings.
10481	GUI_OPERATION_DIALOG__HELP	Help -- The name of the guioperationdialog_help operation used to implement the "Help" button on the dialog box
10482	GUI_OPERATION_DIALOG__YES	Yes -- Alternate label for "OK" button

10483	GUI_OPERATION_DIALOG__NO	No -- Alternate label for "Cancel" button
10501	GUI_OPERATION__CLASSNAME	Operation -- The name of the gui_operation type
10526	GUI_OPERATION_TYPE__CLASSNAME	Operation Type -- The name of the gui_operationtype type
10551	GUI_OPTION__CLASSNAME	Option -- The name of the gui_option type
10576	GUI_PULLDOWN__CLASSNAME	Pulldown Menu -- The name of the gui_pulldown type
10601	GUI_RECTANGLE__CLASSNAME	Rectangle -- The name of the gui_rectangle type
10626	GUI_REFERENCE_FIELD__CLASSNAME	Reference Field -- The name of the gui_referencefield type
10627	GUI_REFERENCE_FIELD__WRONG_TYPE	This field can only hold objects that are of type %s. The message given when an object of the wrong type is dropped in the field
10628	GUI_REFERENCE_FIELD__FIND_FAILED	Could not find an instance of %s which matches the string %s. The message given when the findinstance method returns NULL.
10651	GUI_SCROLLER__CLASSNAME	Scroller -- The name of the gui_scroller type
10676	GUI_SUBSET__CLASSNAME	Subset -- The name of the gui_subset type
10677	GUI_SUBSET__SELECT_ALL	Select All -- The name of the guisubset_selectall operation for setting the subset equal to the set
10678	GUI_SUBSET__UNSELECT_ALL	Unselect All -- The name of the guisubset_unselectall operation for clearing the subset.
10701	GUI_TEXT__CLASSNAME	Text -- The name of the gui_text type
10726	GUI_TEXT_VIEW__CLASSNAME	Option -- The name of the gui_option type



---

10751	GUI_TOGGLE__CLASSNAME	Toggle -- The name of the <code>gui_toggle</code> type
10776	GUI_TYPE__CLASSNAME	Type -- The name of the <code>gui_type</code> type
10801	GUI_VIEW__CLASSNAME	View -- The name of the <code>gui_view</code> type
10802	GUI_VIEW__UNDEFINED_DROP	Dropping an instance of %s on an instance of %s is undefined. By default, this message will be presented for drag and drop
10803	GUI_VIEW__CLOSE	Close -- The name of the <code>guiview_close</code> operation used for closing windows
10816	GUI_VMESSAGE__APPNAME	Versant Message -- The name of the application
10817	GUI_VMESSAGE__USAGE	Usage <code>vmessage &lt;target&gt; &lt;message&gt;</code> -- The usage message
10818	GUI_VMESSAGE__INVALID_TARGET	Invalid target. -- Error printed when <code>sendmessage</code> fails.
10819	GUI_VMESSAGE__MESSAGE_FAILED	Failed to send message -- Error printed when <code>sendmessage</code> fails.
10820	GUI_VMESSAGE__VIEWS _UNAVAILABLE	The DISPLAY environment variable must be set. The error message to give when no X display is specified
10821	GUI_VMESSAGE__MINUS_INFO	-info -- Localized alias for -info
10822	GUI_VMESSAGE__MINUS_RESET	-reset -- Localized alias for -reset
10826	GUI_WINDOW__CLASSNAME	Window -- The name of the <code>gui_window</code> type
10831	GUI_PERCENT_VIEW__CLASSNAME	Percent Bar -- The name of the <code>gui_percentview</code> type
10836	GUI_PERCENT_DIALOG__CLASSNAME	Percent Dialog -- The name of the <code>gui_percentdialog</code> type
10837	GUI_PERCENT_DIALOG__ABORT	Cancel -- The name of the <code>percentdialog_abort</code> option

10841	GUI_TEXT_DIALOG__CLASSNAME	Text Dialog -- The name of the gui_textdialog type
10842	GUI_TEXT_DIALOG__ABORT	Cancel -- The name of the textdialog_abort option

---

## SQL & ST ERRORS, SERIES - 20000

20001	ST_INVALID_PARAMETER	Invalid parameter at: Parameter passed to the given VERSANT defined primitive method is not of the expected class at the indicated position
20002	ST_MISSING_VCLASSINFO	Internal error - vclassinfo object missing. An internal error not currently handled, please contact and report this error to Versant Object Technology Inc.
20003	ST_MISSING_CLASS_AUXINFO	Database class is found missing auxiliary information. Start from 2.0 release, the VERSANT/Smalltalk interface will use a auxiliary dictionary to hold information to be used by the language interface. And it is found to be missing from the database class currently being accessed.
20004	ST_NON_SMALLTALK_RELATED_CLASS	This class is not defined from the Smalltalk/VERSANT interface. It is found that the corresponding database class doesn't have the same language bit set in the auxiliary info dictionary. This indicates that it is not a class defined with the 2.0 release of the VERSANT/Smalltalk interface. The Smalltalk interface would make needed checking to ensure that the a class defined from a pre 2.0 VERSANT/Smalltalk interface would be automatically converted.

20005	ST_CLASS_SIGNATURE_NOT_FOUND	The class is missing class signature. Class signature is not found for the given class. It could mean that the database class is not defined and/or modified through the Smalltalk interface, and hence, you might not be able to access it through the VERSANT/Smalltalk interface.
20006	ST_MEMORY_ALLOCATION_FAILED	Memory allocation in UDP failed. Memory allocation failed in the VERSANT/Smalltalk defined primitive, and it might mean that there is no memory left.
20007	ST_HIERARCHY_SIGNATURE_NOT_FOUND	The class is missing hierarchy signature. The Smalltalk/VERSANT interface would compute a signature for the corresponding database class when it is described to the database, which will be used to find out if the database class has the same inheritance hierarchy as the one the database class is defined from.
20008	ST_SOURCE_CHECKSUM_NOT_FOUND	The database class is missing source check sum. The VERSANT/Smalltalk interface maintains a source check sum value to be used for tell if the source are the same between the database class and that of the Smalltalk class. And it is found that the class does not have this info.

---

20009	ST_STORAGE_TRANSFORM_NOT_FOUND	Database class misses storage transformation encoding. VERSANT/Smalltalk interface maintains storage transformation encoding in the database class to indicate how a given attribute should be translated in the the Smalltalk space. And it is found to be missing. This might indicate the the class is notdefined throught the VERSANT/Smalltalk interface. And you might not be able to access this class from VERSANT/Smalltalk.
20010	ST_CLASS_SOURCE_NOT_FOUND	Database class missing corresponding Smalltalk source. Starting from 2.0 release, database classes defined through the VERSANT/Smalltalk interface will also hold the file-in source of the corresponding Smalltalk class. And that is found to be missing.
20011	ST_CLASS_TYPE_ENCODING_MISSING	Database class missing type encoding for the corresponding ST Class. Starting from 2.0 release, database classes defined through the VERSANT/Smalltalk interface will hold a type encoding to tell if the corresponding Smalltalk class is a variable class.
20012	ST_TIMESTAMP_NOT_FOUND	Database class missing time stamp. When a database class is defined or updated through the VERSANT/Smalltalk interface, a time stamp will be generated and stored in the auxiliarly info dictionary. And it is found missing.

20013	ST_DEFINED_FROM_PRE_2_0_RELEASE	Class defined using 1.x interface. Smalltalk classes defined using 1.x interface do not contain schema information necessary for schema evolution functionality added starting from version 2.x
20014	ST_PREDBLOCK_INVALID	Invalid Predicate specified for query. The conjunctions in a predblock must all have the same type otherwise parsing error at smalltalk level has occurred.
20101	ST_OE_SYS_NO_ERR	No failure code available. Mapped from the Object Engine error for consistent error handling in visualworks Smalltalk interface.
20102	ST_OE_SYS_OUT_OF_RANGE	C argument out of range; probably not large enough data. Mapped from the Object Engine error for consistent error handling in visualworks Smalltalk interface.
20103	ST_OE_SYS_NON_OOP	Oop argument not an oop. Mapped from the Object Engine error for consistent error handling in visualworks Smalltalk interface.
20104	ST_OE_SYS_WRONG_CLASS	Oop argument incorrect type. Mapped from the Object Engine error for consistent error handling in visualworks Smalltalk interface.
20105	ST_OE_SYS_OBJECT_TOO_SMALL	Smalltalk datum too small. Mapped from the Object Engine error for consistent error handling in visualworks Smalltalk interface.

---

20106	ST_OE_SYS_ALLOCATION_FAILED	Smalltalk object memory allocation failed. Mapped from the Object Engine error for consistent error handling in visualworks Smalltalk interface.
20107	ST_OE_SYS_STORE_FAILURE	The function illegally attempted to place an object reference into another object. Mapped from the Object Engine error for consistent error handling in visualworks Smalltalk interface.
20201	SQL_ORACLE_ERROR	Oracle defined error. An Oracle defined error is raised. The corresponding error message string also gives the error number defined by Oracle. Consult Oracle error message and codes manual for detail information.
20202	SQL_INTERNAL_ERROR	Internal error. An internal error not currently handled, please contact and report this error to Versant Object Technology Inc.
20203	SQL_CANNOT_INITIALIZE_CONNECTION	Failed to initialize a database connection. Fail to initialize a database connection. Should check if you have install the interface correctly.
20204	SQL_FAIL_TO_ACCESS_ERROR_MSG	Failed to access the Oracle error message. A call to Oracle C interface function failed and a try in accessing the corresponding Oracle error message is also failed. You might need to check if you have you Oracle product installed and startup correctly.
20205	SQL_OUT_OF_MEMORY	Unable to allocate run-time memory. The interface failed to allocate run-time memory for use by a VERSANT defined primitive function.

20206	SQL_INVALID_PARAMETER	Invalid parameter at: Parameter passed to the given VERSANT defined primitive method is not of the expected class at the indicated position, or the value is invalid.
20207	SQL_INVALID_STATEMENT_INDEX	Invalid SQL statement handle index. The handle instance variable associated with the VSQL instance is invalid. This is normally resulted from an image saving during an active database session. In a new database session, You have create new VSQL statement handle corresponding to newly created database connection.
20208	SQL_INVALID_CONNECTION_INDEX	Invalid SQL database connection handle index. The handle instance variable associated with the vsqldatabase instance is invalid. This is normally resulted from an image saving during an active database session. You might need to recreate a vsqldatabase instance and connect to ORACLE database.



---

20209	SQL_INVALID_DATA_TYPE	Invalid SQL data type used. It is found that a SQL data type is used which is not actually supported in the current release. Please contact and report this error to Versant Object Technology Inc.
20210	SQL_INVALID_COLUMN_NUMBER	Invalid column number encountered. During the process of fetching data records from Oracle database, it is found that the column number associated with the VSQL statement handle is incorrect. This is normally caused by using an obsolete statement cursor.

## VERSANT STATISTICS ERRORS, SERIES - 30000

31000	STAT_FE_BEGIN	DO NOT ADD FRONTEND STATISTICS BEFORE HERE
31001	STAT_FE_NET_READS	Reads from back end
31002	STAT_FE_NET_BYTES_READ	Bytes read from back end
31003	STAT_FE_NET_READ_TIME	Seconds reading from back end
31004	STAT_FE_NET_WRITES	Writes to back end
31005	STAT_FE_NET_BYTES_WRITTEN	Bytes written to back end
31006	STAT_FE_NET_WRITE_TIME	Seconds writing to back end
31007	STAT_FE_SWAPPED	Dirty objects swapped out of object cache
31008	STAT_FE_READS	Objects read into object cache
31009	STAT_FE_VM_MAJ_FAULTS	Virtual memory major page faults
31010	STAT_FE_USER_TIME	Seconds not in OS kernel functions
31011	STAT_FE_SYSTEM_TIME	Seconds in OS kernel functions
31012	STAT_FE_REAL_TIME	Seconds of real time
31013	STAT_FE_WRITES	Objects written from object cache
31014	STAT_FE_LATCH_WAIT_TIME	Seconds waiting for latch
31015	STAT_FE_SWAPPED_DIRTY	Objects written as a result of object swapping
31016	STAT_FE_END	DO NOT ADD FRONTEND STATISTICS AFTER HERE
32000	STAT_SE_BEGIN	DO NOT ADD SESSION STATISTICS BEFORE HERE
32001	STAT_SE_NET_READS	Reads from back end
32002	STAT_SE_NET_BYTES_READ	Bytes read from back end
32003	STAT_SE_NET_READ_TIME	Seconds reading from back end
32004	STAT_SE_NET_WRITES	Writes to back end
32005	STAT_SE_NET_BYTES_WRITTEN	Bytes written to back end
32006	STAT_SE_NET_WRITE_TIME	Seconds writing to back end
32007	STAT_SE_SWAPPED_DIRTY	Objects written as a result of object swapping
32008	STAT_SE_CODS	Cods in object cache
32009	STAT_SE_OBJS	Objects in object cache
32010	STAT_SE_OBJS_DIRTY	Dirty objects in cache

---

32011	STAT_SE_SWAPPED	Objects swapped out of object cache
32012	STAT_SE_READS	Objects read into object cache
32013	STAT_SE_WRITES	Objects written from object cache
32014	STAT_SE_HEAP_USED	Bytes used in front-end heap
32015	STAT_SE_HEAP_FREE	Bytes free in front-end heap
32017	STAT_SE_HEAP_ALLOCATES	Number of "allocate" operations on front-end heap
32016	STAT_SE_HEAP_FREES	Number of "free" operations on front-end heap
32018	STAT_SE_HEAP_SMALL_USED	Bytes used for allocations smaller than 1 page
32019	STAT_SE_HEAP_SMALL_FREE	Bytes free for allocations smaller than 1 page
32020	STAT_SE_HEAP_SMALL_ALLOCATES	Number of "allocate" operations smaller than 1 page
32021	STAT_SE_HEAP_SMALL_FREES	Number of "free" operations smaller than 1 page
32022	STAT_SE_HEAP_MAX_GAP	Size of largest free area in front-end heap in bytes
32023	STAT_SE_HEAP_TOTAL_SEGMENTS	Number of segments in front-end heap
32024	STAT_SE_HEAP_EMPTY_SEGMENTS	Number of empty segments in front-end heap
32025	STAT_SE_LATCH_WAIT_TIME	Seconds waiting for latch
32026	STAT_SE_HEAP_SMALL_ALLOCATES_0	Allocates in size class 0 (1-8 bytes)
32027	STAT_SE_HEAP_SMALL_ALLOCATES_1	Allocates in size class 1 (9-12 bytes)
32029	STAT_SE_HEAP_SMALL_ALLOCATES_3	Allocates in size class 3 (17-20 bytes)
32030	STAT_SE_HEAP_SMALL_ALLOCATES_4	Allocates in size class 4 (21-24 bytes)
32031	STAT_SE_HEAP_SMALL_ALLOCATES_5	Allocates in size class 5 (25-32 bytes)
32032	STAT_SE_HEAP_SMALL_ALLOCATES_6	Allocates in size class 6 (33-40 bytes)
32033	STAT_SE_HEAP_SMALL_ALLOCATES_7	Allocates in size class 7 (41-48 bytes)

32034	STAT_SE_HEAP_SMALL_ALLOCATES_8	Allocates in size class 8 (49-64 bytes)
32035	STAT_SE_HEAP_SMALL_ALLOCATES_9	Allocates in size class 9 (65-80 bytes)
32036	STAT_SE_HEAP_SMALL_ALLOCATES_10	Allocates in size class 10 (81-96 bytes)
32037	STAT_SE_HEAP_SMALL_ALLOCATES_11	Allocates in size class 11 (97-128 bytes)
32038	STAT_SE_HEAP_SMALL_ALLOCATES_12	Allocates in size class 12 (129-160 bytes)
32039	STAT_SE_HEAP_SMALL_ALLOCATES_13	Allocates in size class 13 (161-192 bytes)
32040	STAT_SE_HEAP_SMALL_ALLOCATES_14	Allocates in size class 14 (193-256 bytes)
32041	STAT_SE_HEAP_SMALL_ALLOCATES_15	Allocates in size class 15 (257-336 bytes)
32042	STAT_SE_HEAP_SMALL_ALLOCATES_16	Allocates in size class 16 (337-408 bytes)
32043	STAT_SE_HEAP_SMALL_ALLOCATES_17	Allocates in size class 17 (409-512 bytes)
32044	STAT_SE_HEAP_SMALL_ALLOCATES_18	Allocates in size class 18 (513-680 bytes)
32045	STAT_SE_HEAP_SMALL_ALLOCATES_19	Allocates in size class 19 (681-816 bytes)
32046	STAT_SE_HEAP_SMALL_ALLOCATES_20	Allocates in size class 20 (817-1024 bytes)
32047	STAT_SE_HEAP_SMALL_ALLOCATES_21	Allocates in size class 21 (1025-1360 bytes)
32048	STAT_SE_HEAP_SMALL_ALLOCATES_22	Allocates in size class 22 (1361-1632 bytes)
32049	STAT_SE_HEAP_SMALL_ALLOCATES_23	Allocates in size class 23 (1633-2048 bytes)
32050	STAT_SE_HEAP_SMALL_ALLOCATES_24	Allocates in size class 24 (2049-2720 bytes)
32051	STAT_SE_HEAP_SMALL_ALLOCATES_25	Allocates in size class 25 (2721-3072 bytes)

---

32052	STAT_SE_HEAP_SMALL_ALLOCATES _26	Allocates in size class 26 (3073-4096 bytes)
32053	STAT_SE_HEAP_SMALL_ALLOCATES _27	Allocates in size class 27 (4097-5120 bytes)
32054	STAT_SE_HEAP_SMALL_ALLOCATES _28	Allocates in size class 28 (5121-6144 bytes)
32055	STAT_SE_HEAP_SMALL_ALLOCATES _29	Allocates in size class 29 (6145-10240 bytes)
32056	STAT_SE_HEAP_SMALL_ALLOCATES _30	Allocates in size class 30 (10241-12288 bytes)
32057	STAT_SE_HEAP_SMALL_ALLOCATES _31	Allocates in size class 31 (12289-20480 bytes)
32058	STAT_SE_HEAP_SMALL_FREES_0	Number of frees in size class 0 (1-8 bytes)
32059	STAT_SE_HEAP_SMALL_FREES_1	Number of frees in size class 1 (9-12 bytes)
32060	STAT_SE_HEAP_SMALL_FREES_2	Number of frees in size class 2 (13-16 bytes)
32061	STAT_SE_HEAP_SMALL_FREES_3	Number of frees in size class 3 (17-20 bytes)
32062	STAT_SE_HEAP_SMALL_FREES_4	Number of frees in size class 4 (21-24 bytes)
32063	STAT_SE_HEAP_SMALL_FREES_5	Number of frees in size class 5 (25-32 bytes)
32064	STAT_SE_HEAP_SMALL_FREES_6	Number of frees in size class 6 (33-40 bytes)
32065	STAT_SE_HEAP_SMALL_FREES_7	Number of frees in size class 7 (41-48 bytes)
32066	STAT_SE_HEAP_SMALL_FREES_8	Number of frees in size class 8 (49-64 bytes)
32067	STAT_SE_HEAP_SMALL_FREES_9	Number of frees in size class 9 (65-80 bytes)
32068	STAT_SE_HEAP_SMALL_FREES_10	Number of frees in size class 10 (81-96 bytes)
32069	STAT_SE_HEAP_SMALL_FREES_11	Number of frees in size class 11 (97-128 bytes)
32070	STAT_SE_HEAP_SMALL_FREES_12	Number of frees in size class 12 (129-160 bytes)

32071	STAT_SE_HEAP_SMALL_FREES_13	Number of frees in size class 13 (161-192 bytes)
32072	STAT_SE_HEAP_SMALL_FREES_14	Number of frees in size class 14 (193-256 bytes)
32073	STAT_SE_HEAP_SMALL_FREES_15	Number of frees in size class 15 (257-336 bytes)
32074	STAT_SE_HEAP_SMALL_FREES_16	Number of frees in size class 16 (337-408 bytes)
32075	STAT_SE_HEAP_SMALL_FREES_17	Number of frees in size class 17 (409-512 bytes)
32076	STAT_SE_HEAP_SMALL_FREES_18	Number of frees in size class 18 (513-680 bytes)
32077	STAT_SE_HEAP_SMALL_FREES_19	Number of frees in size class 19 (681-816 bytes)
32078	STAT_SE_HEAP_SMALL_FREES_20	Number of frees in size class 20 (817-1024 bytes)
32079	STAT_SE_HEAP_SMALL_FREES_21	Number of frees in size class 21 (1025-1360 bytes)
32080	STAT_SE_HEAP_SMALL_FREES_22	Number of frees in size class 22 (1361-1632 bytes)
32081	STAT_SE_HEAP_SMALL_FREES_23	Number of frees in size class 23 (1633-2048 bytes)
32082	STAT_SE_HEAP_SMALL_FREES_24	Number of frees in size class 24 (2049-2720 bytes)
32083	STAT_SE_HEAP_SMALL_FREES_25	Number of frees in size class 25 (2721-3072 bytes)
32084	STAT_SE_HEAP_SMALL_FREES_26	Number of frees in size class 26 (3073-4096 bytes)
32085	STAT_SE_HEAP_SMALL_FREES_27	Number of frees in size class 27 (4097-5120 bytes)
32086	STAT_SE_HEAP_SMALL_FREES_28	Number of frees in size class 28 (5121-6144 bytes)
32087	STAT_SE_HEAP_SMALL_FREES_29	Number of frees in size class 29 (6145-10240 bytes)
32088	STAT_SE_HEAP_SMALL_FREES_30	Number of frees in size class 30 (10241-12288 bytes)
32089	STAT_SE_HEAP_SMALL_FREES_31	Number of frees in size class 31 (12289-20480 bytes)
32090	STAT_SE_END	DO NOT ADD SESSION STATISTICS AFTER HERE

---

33000	STAT_BE_BEGIN	DO NOT ADD BACKEND STATISTICS BEFORE HERE
33001	STAT_BE_NET_RPCs	Database rpcs received from cli- ents
33002	STAT_BE_NET_READS	Reads from front end
33003	STAT_BE_NET_BYTES_READ	Bytes read from front end
33004	STAT_BE_NET_READ_TIME	Seconds reading from front end
33005	STAT_BE_NET_WRITES	Writes to front end
33006	STAT_BE_NET_BYTES_WRITTEN	Bytes written to front end
33007	STAT_BE_NET_WRITE_TIME	Seconds writing to front end
33008	STAT_BE_LOCKS_GRANTED	Locks requested and granted
33009	STAT_BE_LOCK_WAITS	Lock waits which occurred
33010	STAT_BE_OBE_LOCKS_WAITING	1 if waiting for lock, 0 other- wise
33011	STAT_BE_LOCK_TIMEOUTS	Timeouts waiting for locks
33012	STAT_BE_LOCK_WAIT_TIME	Seconds clients spent waiting for locks
33013	STAT_BE_LOCK_DEADLOCKS	Deadlocks occurred
33014	STAT_BE_OBJ_SENT	Objects sent to front end
33015	STAT_BE_OBJ_RECEIVED	Objects received from front end
33016	STAT_BE_DATA_READS	Pages read from sysvol + added volumes
33017	STAT_BE_DATA_WRITES	Pages written to sysvol + added volumes
33018	STAT_BE_XACT_STARTED	Transactions started
33019	STAT_BE_XACT_COMMITTED	Transactions committed
33020	STAT_BE_XACT_ROLLED_BACK	Transactions rolled back
33021	STAT_BE_XACT_ACTIVE	Active transactions
33022	STAT_BE_QRY_BTREE_OBJS	Objects read during B-tree query
33023	STAT_BE_QRY_HASH_OBJS	Objects read during hash query
33024	STAT_BE_QRY_SCAN_OBJS	Objects read during sequential scan query
33025	STAT_BE_QRY_BTREE_TIME	Seconds spent in B-tree query
33026	STAT_BE_QRY_HASH_TIME	Seconds spent in hash query
33027	STAT_BE_QRY_SCAN_TIME	Seconds spent in sequential scan query
33028	STAT_BE_DATA_LOCATED	Pages found in database page cache
33029	STAT_BE_VM_MAJ_FAULTS	Virtual memory major page faults

---

33030	STAT_BE_USER_TIME	Seconds not in OS kernel functions
33031	STAT_BE_SYSTEM_TIME	Seconds in OS kernel functions
33032	STAT_BE_REAL_TIME	Seconds elapsed
33033	STAT_BE_EV_DEFINED	Events defined
33034	STAT_BE_EV_SYS_RAISED	System events raised
33035	STAT_BE_EV_SYS_DELIVERED	System events delivered
33036	STAT_BE_EV_USER_RAISED	User events raised
33037	STAT_BE_EV_USER_DELIVERED	User events delivered
33038	STAT_BE_LATCH_RELEASED	Number of latches released of any type
33039	STAT_BE_LATCH_GRANTED	Number of latches granted of any type
33040	STAT_BE_LATCH_GRANTED_SDA	Number of SDA latches granted
33041	STAT_BE_LATCH_GRANTED_HEAP	Number of HEAP latches granted
33042	STAT_BE_LATCH_GRANTED_VOLDEV	Number of VOLDEV latches granted
33043	STAT_BE_LATCH_GRANTED_ST	Number of ST latches granted
33044	STAT_BE_LATCH_GRANTED_SS	Number of SS latches granted
33045	STAT_BE_LATCH_GRANTED_SDHS	Number of SDHS latches granted
33046	STAT_BE_LATCH_GRANTED_SDHS_BKT	Number of SDHS_BKT latches granted
33047	STAT_BE_LATCH_GRANTED_PS	Number of PS latches granted
33048	STAT_BE_LATCH_GRANTED_TR	Number of TR latches granted
33049	STAT_BE_LATCH_GRANTED_EV	Number of EV latches granted
33050	STAT_BE_LATCH_GRANTED_PLOG	Number of PLOG latches granted
33051	STAT_BE_LATCH_GRANTED_LLOG	Number of LLOG latches granted
33052	STAT_BE_LATCH_GRANTED_CP	Number of CP latches granted
33053	STAT_BE_LATCH_GRANTED_CP_WAIT	Number of CP_WAIT latches granted
33054	STAT_BE_LATCH_GRANTED_SCH	Number of SCH latches granted
33055	STAT_BE_LATCH_GRANTED_SCE	Number of SCE latches granted
33056	STAT_BE_LATCH_GRANTED_PHY	Number of PHY latches granted
33057	STAT_BE_LATCH_GRANTED_BF	Number of BF latches granted
33058	STAT_BE_LATCH_GRANTED_BF_BKT	Number of BF_BKT latches granted
33059	STAT_BE_LATCH_GRANTED_BF_FREE	Number of BF_FREE latches granted
33060	STAT_BE_LATCH_GRANTED_L2FILE	Number of L2FILE latches granted
33061	STAT_BE_LATCH_GRANTED_SD	Number of SD latches granted
33062	STAT_BE_LATCH_GRANTED_SC	Number of SC latches granted



---

33063	STAT_BE_LATCH_GRANTED_BF_DIRTY	Number of BF_DIRTY latches granted
33064	STAT_BE_LATCH_GRANTED_LOG_UNIT	Number of LOG_UNIT latches granted
33065	STAT_BE_LATCH_GRANTED_TRE	Number of TRE (transaction entry) latches granted
33066	STAT_BE_LATCH_GRANTED_LOCK	Number of LOCK latches granted
33067	STAT_BE_LATCH_GRANTED_L2FILE _DA	Number of L2FILE_DA latches granted
33068	STAT_BE_LATCH_GRANTED_RES3__	Reserved for future latches
33069	STAT_BE_LATCH_GRANTED_RES4__	Reserved for future latches
33070	STAT_BE_LATCH_GRANTED_RES5__	Reserved for future latches
33071	STAT_BE_LATCH_GRANTED_RES6__	Reserved for future latches
33072	STAT_BE_LATCH_GRANTED_RES7__	Reserved for future latches
33073	STAT_BE_LATCH_GRANTED_RES8__	Reserved for future latches
33074	STAT_BE_LATCH_GRANTED_RES9__	Reserved for future latches
33075	STAT_BE_LATCH_GRANTED_RES10__	Reserved for future latches
33076	STAT_BE_LATCH_WAITS	Number of waits for any latch
33077	STAT_BE_LATCH_WAITS_SDA	Number of waits for SDA latch
33078	STAT_BE_LATCH_WAITS_HEAP	Number of waits for HEAP latch
33079	STAT_BE_LATCH_WAITS_VOLDEV	Number of waits for VOLDEV latch
33080	STAT_BE_LATCH_WAITS_ST	Number of waits for ST latch
33081	STAT_BE_LATCH_WAITS_SS	Number of waits for SS latch
33082	STAT_BE_LATCH_WAITS_SDHS	Number of waits for SDHS latch
33083	STAT_BE_LATCH_WAITS_SDHS_BKT	Number of waits for SDHS_BKT latch
33084	STAT_BE_LATCH_WAITS_PS	Number of waits for PS latch
33085	STAT_BE_LATCH_WAITS_TR	Number of waits for TR latch
33086	STAT_BE_LATCH_WAITS_EV	Number of waits for EV latch
33087	STAT_BE_LATCH_WAITS_PLOG	Number of waits for PLOG latch
33088	STAT_BE_LATCH_WAITS_LLOG	Number of waits for LLOG latch
33089	STAT_BE_LATCH_WAITS_CP	Number of waits for CP latch
33090	STAT_BE_LATCH_WAITS_CP_WAIT	Number of waits for CP_WAIT latch
33091	STAT_BE_LATCH_WAITS_SCH	Number of waits for SCH latch
33092	STAT_BE_LATCH_WAITS_SCE	Number of waits for SCE latch
33093	STAT_BE_LATCH_WAITS_PHY	Number of waits for PHY latch
33094	STAT_BE_LATCH_WAITS_BF	Number of waits for BF latch
33095	STAT_BE_LATCH_WAITS_BF_BKT	Number of waits for BF_BKT latch

---

33096	STAT_BE_LATCH_WAITS_BF_FREE	Number of waits for BF_FREE latch
33097	STAT_BE_LATCH_WAITS_L2FILE	Number of waits for L2FILE latch
33098	STAT_BE_LATCH_WAITS_SD	Number of waits for SD latch
33099	STAT_BE_LATCH_WAITS_SC	Number of waits for SC latch
33100	STAT_BE_LATCH_WAITS_BF_DIRTY	Number of waits for BF_DIRTY latch
33101	STAT_BE_LATCH_WAITS_LOG_UNIT	Number of waits for LOG_UNIT latch
33102	STAT_BE_LATCH_WAITS_TRE	Number of waits for TRE (transaction entry) latch
33103	STAT_BE_LATCH_WAITS_LOCK	Number of waits for LOCK latch
33104	STAT_BE_LATCH_WAITS_L2FILE_DA	Number of waits for L2FILE_DA latch
33105	STAT_BE_LATCH_WAITS_RES3__	Reserved for future latches
33106	STAT_BE_LATCH_WAITS_RES4__	Reserved for future latches
33107	STAT_BE_LATCH_WAITS_RES5__	Reserved for future latches
33108	STAT_BE_LATCH_WAITS_RES6__	Reserved for future latches
33109	STAT_BE_LATCH_WAITS_RES7__	Reserved for future latches
33110	STAT_BE_LATCH_WAITS_RES8__	Reserved for future latches
33111	STAT_BE_LATCH_WAITS_RES9__	Reserved for future latches
33112	STAT_BE_LATCH_WAITS_RES10__	Reserved for future latches
33113	STAT_BE_LATCH_WAIT_TIME	Seconds waiting for any latch
33114	STAT_BE_LATCH_WAIT_TIME_SDA	Seconds waiting for SDA latch
33115	STAT_BE_LATCH_WAIT_TIME_HEAP	Seconds waiting for HEAP latch
33116	STAT_BE_LATCH_WAIT_TIME_VOLDEV	Seconds waiting for VOLDEV latch
33117	STAT_BE_LATCH_WAIT_TIME_ST	Seconds waiting for ST latch
33118	STAT_BE_LATCH_WAIT_TIME_SS	Seconds waiting for SS latch
33119	STAT_BE_LATCH_WAIT_TIME_SDHS	Seconds waiting for SDHS latch
33120	STAT_BE_LATCH_WAIT_TIME_SDHS_BKT	Seconds waiting for SDHS_BKT latch
33121	STAT_BE_LATCH_WAIT_TIME_PS	Seconds waiting for PS latch
33122	STAT_BE_LATCH_WAIT_TIME_TR	Seconds waiting for TR latch
33123	STAT_BE_LATCH_WAIT_TIME_EV	Seconds waiting for EV latch
33124	STAT_BE_LATCH_WAIT_TIME_PLOG	Seconds waiting for PLOG latch
33125	STAT_BE_LATCH_WAIT_TIME_LLOG	Seconds waiting for LLOG latch
33126	STAT_BE_LATCH_WAIT_TIME_CP	Seconds waiting for CP latch
33127	STAT_BE_LATCH_WAIT_TIME_CP_WAIT	Seconds waiting for CP_WAIT latch

---

33128	STAT_BE_LATCH_WAIT_TIME_SCH	Seconds waiting for SCH latch
33129	STAT_BE_LATCH_WAIT_TIME_SCE	Seconds waiting for SCE latch
33130	STAT_BE_LATCH_WAIT_TIME_PHY	Seconds waiting for PHY latch
33131	STAT_BE_LATCH_WAIT_TIME_BF	Seconds waiting for BF latch
33132	STAT_BE_LATCH_WAIT_TIME_BF_BKT	Seconds waiting for BF_BKT latch
33133	STAT_BE_LATCH_WAIT_TIME_BF_FREE	Seconds waiting for BF_FREE latch
33134	STAT_BE_LATCH_WAIT_TIME_L2FILE	Seconds waiting for L2FILE latch
33135	STAT_BE_LATCH_WAIT_TIME_SD	Seconds waiting for SD latch
33136	STAT_BE_LATCH_WAIT_TIME_SC	Seconds waiting for SC latch
33137	STAT_BE_LATCH_WAIT_TIME_BF_DIRTY	Seconds waiting for BF_DIRTY latch
33138	STAT_BE_LATCH_WAIT_TIME_LOG_UNIT	Seconds waiting for LOG_UNIT latch
33139	STAT_BE_LATCH_WAIT_TIME_TRE	Seconds waiting for TRE (transaction entry)
33140	STAT_BE_LATCH_WAIT_TIME_LOCK	Seconds waiting for LOCK
33141	STAT_BE_LATCH_WAIT_TIME_L2FILE_DA	Seconds waiting for L2FILE_DA
33142	STAT_BE_LATCH_WAIT_TIME_RES3__	Reserved for future statistics
33143	STAT_BE_LATCH_WAIT_TIME_RES4__	Reserved for future statistics
33144	STAT_BE_LATCH_WAIT_TIME_RES5__	Reserved for future statistics
33145	STAT_BE_LATCH_WAIT_TIME_RES6__	Reserved for future statistics
33146	STAT_BE_LATCH_WAIT_TIME_RES7__	Reserved for future statistics
33147	STAT_BE_LATCH_WAIT_TIME_RES8__	Reserved for future statistics
33148	STAT_BE_LATCH_WAIT_TIME_RES9__	Reserved for future statistics
33149	STAT_BE_LATCH_WAIT_TIME_RES10__	Reserved for future statistics
33150	STAT_BE_END	DO NOT ADD BACKEND STATISTICS AFTER HERE
34000	STAT_DB_BEGIN	DO NOT ADD DATABASE STATISTICS BEFORE HERE
34001	STAT_DB_DISK_FREE	Bytes of storage available
34002	STAT_DB_DISK_RESERVED	Bytes of storage reserved by classes
34003	STAT_DB_NET_RPCS	Database rpcs received from clients
34004	STAT_DB_NET_READS	Reads from front end
34005	STAT_DB_NET_BYTES_READ	Bytes read from front end

34006	STAT_DB_NET_READ_TIME	Seconds reading from front end
34007	STAT_DB_NET_WRITES	Writes to front end
34008	STAT_DB_NET_BYTES_WRITTEN	Bytes written to front end
34009	STAT_DB_NET_WRITE_TIME	Seconds writing to front end
34010	STAT_DB_LOCKS_GRANTED	Locks requested and granted
34011	STAT_DB_LOCK_WAITS	Lock waits which occurred
34012	STAT_DB_OBE_LOCKS_WAITING	Connections waiting for locks
34013	STAT_DB_LOCK_TIMEOUTS	Timeouts waiting for locks
34014	STAT_DB_LOCK_WAIT_TIME	Seconds clients spent waiting for locks
34015	STAT_DB_LOCK_DEADLOCKS	Deadlocks occurred
34016	STAT_DB_OBJ_SENT	Objects sent to front end
34017	STAT_DB_OBJ_RECEIVED	Objects received from front end
34018	STAT_DB_DATA_READS	Pages read from sysvol + added volumes
34019	STAT_DB_DATA_WRITES	Pages written to sysvol + added volumes
34020	STAT_DB_XACT_STARTED	Transactions started
34021	STAT_DB_XACT_COMMITTED	Transactions committed
34022	STAT_DB_XACT_ROLLED_BACK	Transactions rolled back
34023	STAT_DB_CHECKPOINTS	System checkpoints
34024	STAT_DB_XACT_ACTIVE	Active transactions
34025	STAT_DB_QRY_BTREE _OBJS	Objects read during B-tree query
34026	STAT_DB_QRY_HASH_OBJS	Objects read during hash query
34027	STAT_DB_QRY_SCAN_OBJS	Objects read during sequential scan query
34028	STAT_DB_QRY_BTREE _TIME	Seconds spent in B-tree query
34029	STAT_DB_QRY_HASH_TIME	Seconds spent in hash query
34030	STAT_DB_QRY_SCAN_TIME	Seconds spent in sequential scan query
34031	STAT_DB_DATA_LOCATED	Pages found in database page cache
34032	STAT_DB_EV_DEFINED	Events defined
34033	STAT_DB_EV_SYS_RAISED	System events raised
34034	STAT_DB_EV_SYS_DELIVERED	System events delivered
34035	STAT_DB_EV_USER_RAISED	User events raised

---

34036	STAT_DB_EV_USER_DELIVERED	User events delivered
34037	STAT_DB_BF_LLOG_FLUSHES	Number of writes to logical log
34038	STAT_DB_BF_PLOG_FLUSHES	Number of writes to physical log
34039	STAT_DB_LATCH_RELEASED	Number of latches released of any type
34040	STAT_DB_LATCH_GRANTED	Number of requests for any latch
34041	STAT_DB_LATCH_GRANTED_SDA	Number of SDA latches granted
34042	STAT_DB_LATCH_GRANTED_HEAP	Number of HEAP latches granted
34043	STAT_DB_LATCH_GRANTED_VOLDEV	Number of VOLDEV latches granted
34044	STAT_DB_LATCH_GRANTED_ST	Number of ST latches granted
34045	STAT_DB_LATCH_GRANTED_SS	Number of SS latches granted
34046	STAT_DB_LATCH_GRANTED_SDHS	Number of SDHS latches granted
34047	STAT_DB_LATCH_GRANTED_SDHS_BKT	Number of SDHS_BKT latches granted
34048	STAT_DB_LATCH_GRANTED_PS	Number of PS latches granted
34049	STAT_DB_LATCH_GRANTED_TR	Number of TR latches granted
34050	STAT_DB_LATCH_GRANTED_EV	Number of EV latches granted
34051	STAT_DB_LATCH_GRANTED_PLOG	Number of PLOG latches granted
34052	STAT_DB_LATCH_GRANTED_LLOG	Number of LLOG latches granted
34053	STAT_DB_LATCH_GRANTED_CP	Number of CP latches granted
34054	STAT_DB_LATCH_GRANTED_CP_WAIT	Number of CP_WAIT latches granted
34055	STAT_DB_LATCH_GRANTED_SCH	Number of SCH latches granted
34056	STAT_DB_LATCH_GRANTED_SCE	Number of SCE latches granted
34057	STAT_DB_LATCH_GRANTED_PHY	Number of PHY latches granted
34058	STAT_DB_LATCH_GRANTED_BF	Number of BF latches granted
34059	STAT_DB_LATCH_GRANTED_BF_BKT	Number of BF_BKT latches granted
34060	STAT_DB_LATCH_GRANTED_BF_FREE	Number of BF_FREE latches granted
34061	STAT_DB_LATCH_GRANTED_L2FILE	Number of L2FILE latches granted
34062	STAT_DB_LATCH_GRANTED_SD	Number of SD latches granted
34063	STAT_DB_LATCH_GRANTED_SC	Number of SC latches granted
34064	STAT_DB_LATCH_GRANTED_BF_DIRTY	Number of BF_DIRTY latches granted
34065	STAT_DB_LATCH_GRANTED_LOG_UNIT	Number of LOG_UNIT latches granted
34066	STAT_DB_LATCH_GRANTED_TRE	Number of TRE (transaction entry) latches granted
34067	STAT_DB_LATCH_GRANTED_LOCK	Number of LOCK latches granted
34068	STAT_DB_LATCH_GRANTED_L2FILE_D A	Number of L2FILE_DA latches granted

34069	STAT_DB_LATCH_GRANTED_RES3__	Reserved for future latches
34070	STAT_DB_LATCH_GRANTED_RES4__	Reserved for future latches
34071	STAT_DB_LATCH_GRANTED_RES5__	Reserved for future latches
34072	STAT_DB_LATCH_GRANTED_RES6__	Reserved for future latches
34073	STAT_DB_LATCH_GRANTED_RES7__	Reserved for future latches
34074	STAT_DB_LATCH_GRANTED_RES8__	Reserved for future latches
34075	STAT_DB_LATCH_GRANTED_RES9__	Reserved for future latches
34076	STAT_DB_LATCH_GRANTED_RES10__	Reserved for future latches
34077	STAT_DB_LATCH_WAITS	Number of waits for any latch
34078	STAT_DB_LATCH_WAITS_SDA	Number of waits for SDA latch
34079	STAT_DB_LATCH_WAITS_HEAP	Number of waits for HEAP latch
34080	STAT_DB_LATCH_WAITS_VOLDEV	Number of waits for VOLDEV latch
34081	STAT_DB_LATCH_WAITS_ST	Number of waits for ST latch
34082	STAT_DB_LATCH_WAITS_SS	Number of waits for SS latch
34083	STAT_DB_LATCH_WAITS_SDHS	Number of waits for SDHS latch
34084	STAT_DB_LATCH_WAITS_SDHS_BKT	Number of waits for SDHS_BKT latch
34085	STAT_DB_LATCH_WAITS_PS	Number of waits for PS latch
34086	STAT_DB_LATCH_WAITS_TR	Number of waits for TR latch
34087	STAT_DB_LATCH_WAITS_EV	Number of waits for EV latch
34088	STAT_DB_LATCH_WAITS_PLOG	Number of waits for PLOG latch
34089	STAT_DB_LATCH_WAITS_LLOG	Number of waits for LLOG latch
34090	STAT_DB_LATCH_WAITS_CP	Number of waits for CP latch
34091	STAT_DB_LATCH_WAITS_CP_WAIT	Number of waits for CP_WAIT latch
34092	STAT_DB_LATCH_WAITS_SCH	Number of waits for SCH latch
34093	STAT_DB_LATCH_WAITS_SCE	Number of waits for SCE latch
34094	STAT_DB_LATCH_WAITS_PHY	Number of waits for PHY latch
34095	STAT_DB_LATCH_WAITS_BF	Number of waits for BF latch
34096	STAT_DB_LATCH_WAITS_BF_BKT	Number of waits for BF_BKT latch
34097	STAT_DB_LATCH_WAITS_BF_FREE	Number of waits for BF_FREE latch
34098	STAT_DB_LATCH_WAITS_L2FILE	Number of waits for L2FILE latch
34099	STAT_DB_LATCH_WAITS_SD	Number of waits for SD latch
34100	STAT_DB_LATCH_WAITS_SC	Number of waits for SC latch
34101	STAT_DB_LATCH_WAITS_BF_DIRTY	Number of waits for BF_DIRTY latch
34102	STAT_DB_LATCH_WAITS_LOG_UNIT	Number of waits for LOG_UNIT latch

---

34103	STAT_DB_LATCH_WAITS_TRE	Number of waits for TRE (transaction entry) latch
34104	STAT_DB_LATCH_WAITS_LOCK	Number of waits for LOCK latch
34105	STAT_DB_LATCH_WAITS_L2FILE_DA	Number of waits for L2FILE_DA latch
34106	STAT_DB_LATCH_WAITS_RES3__	Reserved for future latches
34107	STAT_DB_LATCH_WAITS_RES4__	Reserved for future latches
34108	STAT_DB_LATCH_WAITS_RES5__	Reserved for future latches
34109	STAT_DB_LATCH_WAITS_RES6__	Reserved for future latches
34110	STAT_DB_LATCH_WAITS_RES7__	Reserved for future latches
34111	STAT_DB_LATCH_WAITS_RES8__	Reserved for future latches
34112	STAT_DB_LATCH_WAITS_RES9__	Reserved for future latches
34113	STAT_DB_LATCH_WAITS_RES10__	Reserved for future latches
34114	STAT_DB_LATCH_WAIT_TIME	Seconds waiting for any latch
34115	STAT_DB_LATCH_WAIT_TIME_SDA	Seconds waiting for SDA latch
34116	STAT_DB_LATCH_WAIT_TIME_HEAP	Seconds waiting for HEAP latch
34117	STAT_DB_LATCH_WAIT_TIME_VOLDEV	Seconds waiting for VOLDEV latch
34118	STAT_DB_LATCH_WAIT_TIME_ST	Seconds waiting for ST latch
34119	STAT_DB_LATCH_WAIT_TIME_SS	Seconds waiting for SS latch
34120	STAT_DB_LATCH_WAIT_TIME_SDHS	Seconds waiting for SDHS latch
34121	STAT_DB_LATCH_WAIT_TIME_SDHS_BKT	Seconds waiting for SDHS_BKT latch
34122	STAT_DB_LATCH_WAIT_TIME_PS	Seconds waiting for PS latch
34123	STAT_DB_LATCH_WAIT_TIME_TR	Seconds waiting for TR latch
34124	STAT_DB_LATCH_WAIT_TIME_EV	Seconds waiting for EV latch
34125	STAT_DB_LATCH_WAIT_TIME_PLOG	Seconds waiting for PLOG latch
34126	STAT_DB_LATCH_WAIT_TIME_LLOG	Seconds waiting for LLOG latch
34127	STAT_DB_LATCH_WAIT_TIME_CP	Seconds waiting for CP latch
34128	STAT_DB_LATCH_WAIT_TIME_CP_WAIT	Seconds waiting for CP_WAIT latch
34129	STAT_DB_LATCH_WAIT_TIME_SCH	Seconds waiting for SCH latch
34130	STAT_DB_LATCH_WAIT_TIME_SCE	Seconds waiting for SCE latch
34131	STAT_DB_LATCH_WAIT_TIME_PHY	Seconds waiting for PHY latch
34132	STAT_DB_LATCH_WAIT_TIME_BF	Seconds waiting for BF latch
34133	STAT_DB_LATCH_WAIT_TIME_BF_BKT	Seconds waiting for BF_BKT latch
34134	STAT_DB_LATCH_WAIT_TIME_BF_FREE	Seconds waiting for BF_FREE latch
34135	STAT_DB_LATCH_WAIT_TIME_L2FILE	Seconds waiting for L2FILE latch

---

34136	STAT_DB_LATCH_WAIT_TIME_SD	Seconds waiting for SD latch
34137	STAT_DB_LATCH_WAIT_TIME_SC	Seconds waiting for SC latch
34138	STAT_DB_LATCH_WAIT_TIME_BF _DIRTY	Seconds waiting for BF_DIRTY latch
34139	STAT_DB_LATCH_WAIT_TIME_LOG _UNIT	Seconds waiting for LOG_UNIT latch
34140	STAT_DB_LATCH_WAIT_TIME_TRE	Seconds waiting for TRE (transac- tion entry)
34141	STAT_DB_LATCH_WAIT_TIME_LOCK	Seconds waiting for LOCK
34142	STAT_DB_LATCH_WAIT_TIME_L2FILE _DA	Seconds waiting for L2FILE_DA
34143	STAT_DB_LATCH_WAIT_TIME_RES3__	Reserved for future statistics
34144	STAT_DB_LATCH_WAIT_TIME_RES4__	Reserved for future statistics
34145	STAT_DB_LATCH_WAIT_TIME_RES5__	Reserved for future statistics
34146	STAT_DB_LATCH_WAIT_TIME_RES6__	Reserved for future statistics
34147	STAT_DB_LATCH_WAIT_TIME_RES7__	Reserved for future statistics
34148	STAT_DB_LATCH_WAIT_TIME_RES8__	Reserved for future statistics
34149	STAT_DB_LATCH_WAIT_TIME_RES9__	Reserved for future statistics
34150	STAT_DB_LATCH_WAIT_TIME _RES10__	Reserved for future statistics
34151	STAT_DB_BF_LLOG_BYTES_WRITTEN	Bytes written to logical log
34152	STAT_DB_BF_PLOG_BYTES_WRITTEN	Bytes written to physical log
34153	STAT_DB_BF_LLOG_FULL	Number of logical log buffer full encountered
34154	STAT_DB_BF_LLOG_END	Number of logical log buffer end encountered
34155	STAT_DB_BF_PLOG_FULL	Number of physical log buffer full encountered
34156	STAT_DB_BF_PLOG_END	Number of physical log buffer end encountered
34157	STAT_DB_AT_ROOT_READ	Number of AT root and AT root bucket pages read
34158	STAT_DB_AT_LEAF_READ	Number of AT leaf pages read
34159	STAT_DB_AT_ROOT_LOCATED	Number of AT root and AT root bucket located
34160	STAT_DB_AT_LEAF_LOCATED	Number of AT leaf pages located
34161	STAT_DB_END	Do not add database statistics after here



---

19501	EXP_YACC_STACK	Parser stack overflow
19502	EXP_SYNTAX_ERROR	line %d: Syntax error, near ``%s''
19503	EXP_MISSING_STAR_LP	Missing ``*）」
19504	EXP_MISSING_STAR_LP_FROM	Missing ``*）」 from line %d
19505	EXP_UNEXPECTED_EOF	Unexpected end-of-file %s from line %d
19506	EXP_WARN_CPP_RESWORD	line %d: Warning, C++ reserved word ``%s''
19507	EXP_WARN_RESWORD	line %d: Warning, Unsupported EXPRESS reserved word ``%s''
19508	EXP_REDEFINED_SYMBOL	line %d: Symbol ``%s'' is rede- fined
19509	EXP_UNRESOLVED_REF	line %d: Unresolved symbol ``%s''
19510	EXP_NON_ABBREVIATED	The ``-a'' option was specified for exp2cxx. However, all of the ENTITY names were short enough. Therefore the corresponding gen- erated class name did not require any abbreviation definitions in th ``.a'' file.
19511	EXP_INVALID_OPTION	Invalid compiler option ``%s''
19512	EXP_OPEN_FILE	Cannot open ``%s'' file
19513	EXP_OUT_OF_MEMORY	System Error
19514	EXP_INIT_ENTITY_ITERATOR	System Error [ used by ` initEntityItera- tor()']
19515	EXP_INIT_ATTR_ITERATOR	System Error [ used by ` initAttrItera- tor()']
19516	EXP_INIT_TYPEDEF_ITERATOR	System Error [ used by ` initTypeDefItera- tor()']
19517	EXP_INIT_SYMBOL_ITERATOR	System Error [ used by ` initSymbolItera- tor()']
19518	EXP_INIT_AGGR_ITERATOR	System Error [ used by ` initAggrItera- tor()']

12026	VHELP_BROKER__WAITING_FOR _SERVER	Waiting for help server to start...This message appears if the server hasn't registered itself within a few seconds after it was started
12027	VHELP_BROKER__TIMEOUT	Time-out while waiting for help server to start.  This message appears if after waiting a few more seconds it still is not registered.
12028	VHELP_BROKER__EXEC_FAILED	Could not execute vhelp. Check your installation.This message appears if execvp fails to execute vhelp.
12051	VHELP_HELP__CLASSNAME	Topic -- The name of the Help type
12052	VHELP_HELP__TTY_FLAG	-tty -- A localized alias for the -tty flag
12053	VHELP_HELP__RESET_FLAG	-reset -- A localized alias for the -reset flag
12054	VHELP_HELP__RESET_MESSAGE	Reseting help server entry in registry.  Message displayed during a reset.
12055	VHELP_HELP__MATCHES_RELATION	Matches... The name of the relationship used to connect a keyword search with topics that matched the keyword.
12056	VHELP_HELP__SEARCHES_RELATION	Searches... The name of the relationship used to connect the general keyword search topic with all the keyword searches performed.
12057	VHELP_HELP__TTY_TOPIC_NAME _FORMAT	Topic Name: %s -- The format used for printing the name of a topic when -tty is used.
12058	VHELP_HELP__TTY_SEE_ALSO _FORMAT	See Also: The text used for labeling the "see also" list of adjacent topics when in -tty mode.

---

12059	VHELP_HELP__TTY_TOPIC_DIVIDER	----- The text used for separating and terminating topics while in -tty mode.
12060	VHELP_HELP__BAD_TOPIC_FILE	The file %s could not be opened. The text displayed when a topic's file cannot be opened.
12061	VHELP_HELP__NO_LIB_VHELP	Could not find %s. The text displayed when \$(OSCROOT)/lib/vhelp can't be found.
12062	VHELP_HELP__NON_EXISTENT_TOPIC	Received message to display non-existent topic %s.  The text displayed when a message to display a non-existent topic is received.
12063	VHELP_HELP__KEYWORD_SEARCH	Keyword Search -- The name of the help_keywordSearch operation
12064	VHELP_HELP_KEYWORD_SEARCH__SEARCH_FOR	Search for: The name of the first argument -- the keyword to search for.
12065	VHELP_HELP_KEYWORD_SEARCH__NO_MATCH	No topics match %s. The message presented if no matches are found.
12066	VHELP_HELP__KEYWORD_TITLE	Keyword Search on %s. The title of the text for automatically generated search topics
12067	VHELP_HELP__KEYWORD_BODY	This topic was automatically generated by a keyword search on %s. The main body the text for automatically generated search topics
12076	VHELP_HELP_VIEW__CLASSNAME	Topic View -- The name of the HelpView type
12077	VHELP_HELP_VIEW__PATH_TEXT	Navigation -- The text used when labeling the embedded path
12078	VHELP_HELP_VIEW__WINDOW_TITLE	Versant Help: Topic %s. -- The title of a Help Topic Window
12079	VHELP_HELP_VIEW__PREVIOUS_TOPIC	Previous -- The title of the HelpView_previousTopic operation
12080	VHELP_HELP_VIEW__NEW_PATH	New Navigation Window -- The title of the HelpView_newPath operation

12081	VHELP_HELP_VIEW__GOTO _SELECTION	Go To Selection -- The title of the HelpView_gotoSelection operation (associated with the text)
12082	VHELP_HELP_VIEW__SEARCH_ON _SELECTION	Keyword Search On Selection -- The title of the HelpView_searchOnSelection operation (associated with the text)
12083	VHELP_HELP_VIEW_SEARCH_ON _SELECTION_NO_MATCH	No topics match %s. -- This error is given when no match is found.
12084	VHELP_HELP_VIEW__TEXT_NAME	Help Text -- The name of the text
12101	VHELP_MAIN__APP_NAME	Versant Help -- The name of the application
12102	VHELP_MAIN__NO_TOPIC	No help available on %s. This error generated when no help topic is found.
12126	VHELP_NODE__CLASSNAME	Node -- The name of the Node type
12151	VHELP_PATH_VIEW__CLASSNAME	Navigation -- The name of the PathView type
12152	VHELP_PATH_VIEW__WINDOW_TITLE	Versant Help: Navigation from %s -- The title of a Navigation Window
12153	VHELP_PATH_VIEW__PREVIOUS_PATH	Return To Previous State -- The name of the PathView_previousPath operation
12154	VHELP_PATH_VIEW__GOTO	Go To Selection -- The name of the PathView_goto operation
12155	VHELP_PATH_VIEW__VIEW	View Selection -- The name of the PathView_view operation
12156	VHELP_PATH_VIEW__LIST_NAME	Related Topics List -- The name of the lists in a PathView
12157	VHELP_PATH_VIEW__LIST_LABEL	Topics related to\n%s: -- The label for navigation list

---

This Chapter explains the support for Microsoft Cluster.

This Chapter covers the following in details:

- Support for Microsoft Cluster Server (MSCS) on Windows (Wolfpack)

## SUPPORT FOR MICROSOFT CLUSTER SERVER (MSCS) ON WINDOWS (WOLFPACK)

Versant supports Wolfpack on Windows Enterprise Edition.

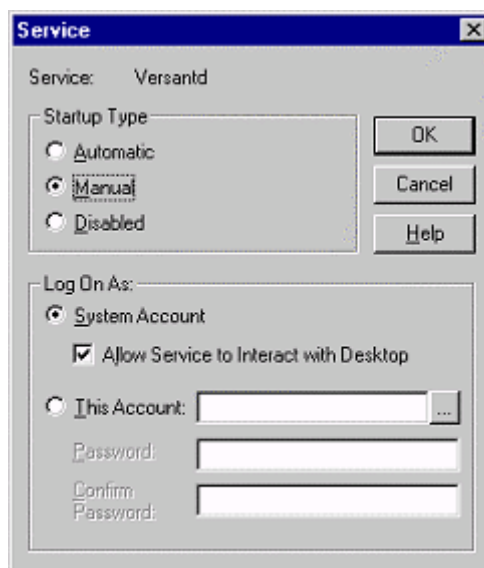
This feature allows you to designate one node in the cluster as a Versant database server and failover automatically to another node if the system crashes. Versant database sessions, which are active at the time of failure, will be lost and their transactions will be rolled back, but new database connections will succeed and will be serviced by the secondary node.

### Versant MSCS Configuration Guide

#### Install Versant on nodes

Install Versant on both the nodes on a local disk drive say `D:\VERSANT` and let the installation pick the default directories for the databases. These entries need to be modified so that Versant can work with MSCS transparently.

After installing Versant and before rebooting the machines, the **Startup Type** for the Versantd service needs to be changed from **Automatic** to **Manual**. To change this setting bring up the services dialog box from the **Control Panel** and double click on Versantd and change the setting as shown below.



## Using Cluster Administrator

Cluster Administrator shows you information about the groups and resources on all of your clusters and specific information about the clusters themselves. A copy of Cluster Administrator is automatically installed on both cluster nodes when you install MSCS. For remote administration, you can install separate copies of Cluster Administrator on other computers on your network running Service Pack 3 and either Windows Workstation or Windows Server version 4.0. The remote and local copies of Cluster Administrator are identical.

## Viewing the Default Groups

Every new MSCS cluster includes two types of default resource groups: Cluster Group and Disk Group. These groups contain the settings for the default cluster and some typical resources that provide generic information and failover policies.

### Cluster Group

The default Cluster Group contains an IP Address resource, a Cluster Name resource, and a Time Service resource. (This group is essential for connectivity to the cluster).

## Disk Group

One Disk Group is created for each disk resource on the shared SCSI bus. A Physical Disk resource is included in each group.

Do not delete or rename the Cluster Group. Instead, model your new groups on the Cluster Group by modifying the Disk Groups or creating a new group as we will see in the following sections.

**NOTE:-** Each cluster needs only one Time Service resource. You do not need, and should not create, a Time Service resource in each group.

## Creating a New Versant Group

Use the New Group wizard in Cluster Administrator to add a new group to your cluster. To start the New Group wizard, on the **File** menu, click **New**, and then click **Group**. For step-by-step instructions, see Cluster Administrator Help. When you add a new group, the New Group wizard guides you through the two-step process. Before running the New Group wizard, make sure you have all the information you need to complete the wizard. Use the following table to prepare to run the wizard.

### Information required to run the New Group wizard

Information required	What it is used for
The name you will assign to the group	The name that you give the group is used only for administrative purposes. It is not the same as the Network Name, which allows users to access resources through Virtual Servers.
The text you will use to describe the group	The group description appears in the right pane of Cluster Administrator when you select the <b>Groups</b> folder in the left pane.
The name of the node that will be the preferred owner of the new group	The preferred owner is the node on which you prefer each group to run.

For a Versant database group you will need the following resources to be part of this new group,



---

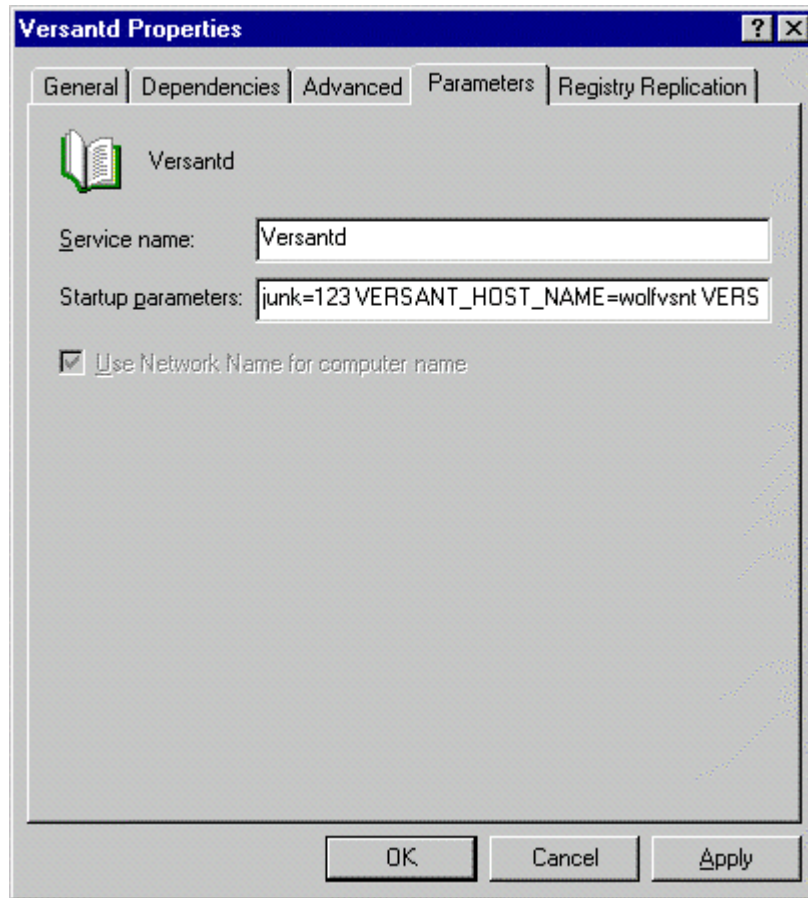
### Information required when adding resources

Resource Type	Specific information you must supply
Physical Disk	Drive letter
IP Address	The IP address (x.x.x.x) and subnet mask (x.x.x.x)
Network Name	The computer name you want to create
Generic Service	Name of the service (Versantd)

## Generic Service

When you add a service to the MSCS clustered environment, click **Generic Service** as the resource type.

After you configure the possible owners and the dependencies, you must supply the name of the service. For example, the Figure below shows the Versantd service name: **Versantd**. You must type the exact name of the service because the method of maintenance of services is specific to the name. The service name is not case sensitive. Specify the startup parameters for the service.



By default, the resource is offline, so you must bring the service online before it successfully operates as an MSCS resource.

An example of startup parameters which need to be provided is given below.

If,

Physical Disk        = X:  
IP Address            = 10.254.3.33  
Network Name = wolfvsnt

And, desired database directory is X:\DB

---

Then, Startup parameters should be as follows for the Versantd service

```
VERSANT_HOST_NAME=wolfvsnt VERSANT_IP=10.254.3.33  
HOMEDRIVE=X: HOMEPATH=\db\ VERSANT_DB=X:\DB  
VERSANT_DBID=X:\DB VERSANT_DBID_NODE=wolfvsnt
```

**IMPORTANT NOTE:-** A known bug occurs while the startup parameters are passed on to the service. As a workaround add an extra parameter at the start and end of the parameter list say **junk=123**

## Host File Changes

Add a line in the hosts file on both nodes in C:\WINNT\system32\drivers\etc\hosts as shown below,

```
10.254.3.33 wolfvsnt
```

## Creating New osc-dbid File

You need to create a new osc-dbid file on the shared disk before using the configuration, to create this file run dbid -N on one of the MSCS nodes which owns the physical disk.

### Adding resource type to the cluster

To add a new resource type to the cluster, use the cluster.exe

```
cluster resourcetype "vsnt_res" /create/dllname:vsnt_res.dll
```

If you don't specify a path to your resource DLL, Cluster Server will search for it first in the cluster directory and then on the system path.

You can now create resources of type "My vsnt\_res" using the cluster.exe, using the following command:

```
cluster resource "My vsnt_res" /create/group:"Group  
Name"/type:"vsnt_res"
```

where:

"Group Name" is the name of the group you want to add to this resource type

The other resource types in "Group Name" should be:

"IP address"  
"Network Name" and  
"Physical Disk".

Install Versant on both nodes, with the Database directory on a shared disk.

## Setting up Versantd Service

Once both the nodes come up, the Versant service "versantd" will be started on both nodes. Verify this by running "net start" on both nodes. A single entry for versantd should be in the list of started services.

On the secondary node (Failover node), stop the versantd service by running "net stop versantd". The Wolfpack resource DLL will now take care of failing over to the secondary node and starting versantd automatically.

When a node is taken offline it will automatically stop the versantd service on that node.

The failback policy should be turned off so that once failover takes place, it should not again failback to the same node.

# Index

## Symbols

.dbtype 302  
 .lock 300, 301  
 .oscxyyzz 304  
 .sharemem 301  
 .systrace 303  
 .vstatsrc 305

## A

access modes 35  
 add database user 182  
 add database volume 135  
 addvol 135  
 alias 74  
 application process parameters 72  
 application process profile file 296  
 archive errors, 52005399 409  
 assertion\_level 119  
 async\_buffer\_cleaner 95  
 async\_logger 95  
 authentication parameter 123  
 authentication\_program 123

## B

backup database 222, 224  
 backup database and start polling 225  
 backup device 236  
 backup level 235  
 backward compatibility 29  
 be\_syslog\_level 105  
 bf\_dirty\_high\_water\_mark 108  
 bf\_dirty\_low\_water\_mark 108  
 block signal 79

## C

C++ libraries 291  
 checking 339, 360  
 class 109  
 class, create index 155  
 class, create partition 155  
 class, create schema file 211  
 class, create unique attribute 155  
 class, create unique index 155  
 class, delete class 185  
 class, delete class and instances 185  
 class, delete index 155

class, delete instances 186  
 class, delete unique attribute 155  
 clean up application process resources 138  
 clean up server process resources 138  
 cleanbe 138  
 cleanfe 138  
 cnvrtdb 139  
 command 218  
 commit\_flush 98  
 compardb 138  
 compare databases 138  
 comparedb 138  
 configuration files 303  
 configuration parameters for UNIX 318  
 configuration parameters for Windows 351  
 configuration parameters overview 316  
 connect 75  
 connect database 75  
 Container errors, 62006399 419  
 conversion of VERSANT database to release  
     7.0 142  
 convert database 139  
 convertdb 139  
 copy database 245, 248  
 create a database 99  
 create database directories and files 193  
 create database system identifier file 149  
 create index 155  
 create partition 155  
 create replica database 145  
 create schema description file 211  
 create schema implementation file 211  
 create the database 67  
 create unique attribute 155  
 create unique index 155  
 createreplica 145  
 creating a new osc-dbid file 569  
 creating a new VERSANT group 566  
 creatrep 145  
 custom\_be\_plugins 98  
 customized user authentication 29

## D

database administrator 54  
 database administrator authentication 21  
 database administrator dba 15  
 database application process, clean up  
     resources 138  
 database application profile, file 296

- database application profile, set database alias 74
- database application profile, set heap increment 81
- database application profile, set heap size 80
- database application profile, signal block 79
- database components 56
- database creation parameters 89
- database creation procedures 60
- database directories 293
- database file, backup file 301
- database file, database type 302
- database file, lock file 300
- database file, logical log volume 295
- database file, machine configuration 304
- database file, password file 301
- database file, shared memory 301
- database lock file 300, 301
- database parameter 90
- database process profile file 300
- database server process, clean up resources 138
- database server profile, auto addvol 95
- database server profile, auto addvol aggression 97
- database server profile, auto addvol threshold 96
- database server profile, file 300
- database server profile, logical log size and path 90
- database server profile, set buffer clean up 108
- database server profile, set buffer cleanup thread 95
- database server profile, set commit delete 98
- database server profile, set commit flush 98
- database server profile, set data volume location 99
- database server profile, set data volume size 99
- database server profile, set extent size 89
- database server profile, set heap size 110
- database server profile, set logging cleanup thread 95
- database server profile, set logging off 101
- database server profile, set logging on 101
- database server profile, set logical log buffer size 111
- database server profile, set logical log volume size and path 90, 91
- database server profile, set max number of page buffers 113
- database server profile, set physical log buffer size 115
- database server profile, set system volume size and path 92
- database shared memory file 301
- database space parameter 126
- database space parameters 88, 126
- database system administrator dbsa 14
- database system identifier file 308
- database system identifier file, oscdbid 55
- database type file 302
- database user authentication 25
- database users (dbuser) 16
- database volume, logical log 295
- database volume, physical log volume 295
- database volume, system volume 295
- Database Volumes 56
- database volumes and files 294
- database, add user 182
- database, add volume 135
- database, backup database 222
- database, compare databases 138
- database, connect database 75
- database, convert database 139
- database, copy database 245, 248
- database, create a data volume 99
- database, create files 193
- database, create partition 155
- database, create system identifier file 149
- database, database owner 54
- database, display database contents 148
- database, export database contents 269
- database, get backup information 222
- database, get database mode 150
- database, get database users 182
- database, import database contents 269
- database, remove database 204
- database, remove user 182
- database, restore database 222
- database, set database identifier 217
- database, set database mode 150
- database, set database timeout 109
- database, set default volume 155
- database, set latch number 114
- database, start database 219
- database, stop database 220

database, synchronize two databases 204  
 database, system information file 304  
 database\_directory 294  
 database\_name 296  
 database\_root 293  
 datavol 99  
 db\_timeout 109  
 db2tty 148  
 dba utility logging 298  
 dbconnect\_timeout 79  
 dbid 149  
 dbid -n 569  
 dbinfo 150  
 dblist 154  
 dbtool 155  
 dbuser 182  
 delete class 185  
 delete index 155  
 delete replica database 206  
 delete unique attribute 155  
 demonstration programs 287  
 derived statistics file 305  
 directories and files 282  
 directories and files, all installations 308  
 display database contents 148  
 drop class and instances 185  
 drop class instances 186  
 dropclass 185  
 dropcls 185  
 dropinst 186

## E

environment variable 322  
 environment variable, set authentication program  
     path 123  
 error message file 297  
 error, get error code information 250  
 error, index error message file 251  
 error, message file 297  
 estimated\_connections 80  
 estimated\_objects 80  
 etc 356  
 event daemon name 124  
 event notification errors series 6500 421  
 event notification parameters 124  
 event notification, event daemon name 124  
 event notification, set persistence of  
     registrations 125  
 event\_daemon 124

event\_daemon\_notification 126  
 event\_msg\_transient\_queue\_size 125  
 event\_registration\_mode 125  
 example 178, 218  
 export database contents 269  
 extent\_size 89

## F

fault tolerant server, create replica database 145  
 fault tolerant server, delete replica database 206  
 fault tolerant server, set synchronization off 188  
 fault tolerant server, set synchronization on 188  
 for nt 295, 302, 303  
 for UNIX 294, 302, 303, 306  
 ftstool 188  
 functional parameters 74, 95

## G

general executables 286  
 general usage notes 219  
 generic service 567  
 genericobject 76  
 get backup file information 222  
 get connection information 155, 255  
 get database information 155  
 get database list 154  
 get database mode 150  
 get database users 182  
 get defined statistics 255  
 get environment information 203  
 get error code information 250  
 get lock information 155, 255  
 get object information 155  
 get process information 155  
 get statistics from file 255  
 get statistics from memory 255  
 get storage volume information 155  
 get thread information 155  
 get transaction information 155, 255

## H

heap\_size (application) 80  
 heap\_size (database) 110  
 heap\_size\_increment (application) 81  
 heap\_size\_increment (database) 110  
 host file changes 569  
 hostname 356

## I

- import database contents 269
- index 111
- index error message file 251
- install VERSANT on nodes 564

## J

- java libraries 292
- jre directory 291

## L

- large file support 92
- license file 312
- license.xml 312
- listening on multiple ports 198
- llog\_buf\_size 111
- llogvol 90
- lock, short, set short lock wait timeout 112
- lock, short, set short locking off 101
- lock, short, set short locking on 101
- lock\_batch\_size 112
- lock\_wait\_timeout 112
- locking 101
- locking errors, 50005199 407
- log 218
- LOGFILE 297
- logging 101
- logical log volume 295
- logical.log 295
- loose\_schema\_mapping 76

## M

- machine configuration file 304, 323
- makedb 193
- max\_objects 81
- max\_page\_bufs 113
- memory, clean up application process resources 138
- memory, clean up server process resources 138
- memory, set cache swap threshold 82
- microsoft cluster server (mscs) for windows (wolfpack) 564
- move a database 69
- move a database volume 99
- multi\_latch 114
- multiple VERSANT releases on same machine 198

## N

- NetworkServices 198
- notes 254

## O

- o\_indextype 111
- obe\_port\_begin 114
- obe\_port\_end 114
- optimal automatic add volume settings 97
- optionally edit database profile 64
- oscp 203
- out\_of\_space\_warning\_theshold\_increment 127
- out\_of\_space\_warning\_threshold 126

## P

- partition 155
- passing parameters to remote machines 321, 351
- password 264
- path 324, 352
- personal.flg group.flg 302
- physical log volume 295
- physical.log 295
- plog\_buf\_size 115
- plogvol 91
- polling 204
- polling\_optimize 115
- port 218
- profile.be 300
- public access 35

## Q

- query errors, 54005599 411

## R

- remove database 204
- remove database user 182
- removedb 204
- removereplica 206
- removrep 206
- reorgdb 207
- restore and rename database 228
- restore database 222, 226
- roll forward, set roll forward on 222
- roll forward, start roll forward archiving 222
- roll forward, stop roll forward archiving 222
- root and system directories 285



## S

- ul style="list-style-type: none; padding-left: 0;">
- sch2db() 207
- schcomp 211
- schema errors, 60006199 416
- search order 320
- server process parameters 84
- server process profile kinds of parameters 85
- server process profile location 84
- set application heap increment 81
- set application heap size 80
- set asynchronous database cleanup 95
- set asynchronous logging cleanup 95
- set buffer clean up high water mark 108
- set buffer clean up low water mark 108
- set cache swap threshold 82
- set database alias 74
- set database heap size 110
- set database identifier 217
- set database mode 150
- set database timeout 109
- set default storage volume for a class 155
- set flush after commit 98
- set location of user authentication program 123
- set logging off 101
- set logging on 101
- set logical log volume size 90
- set logical logging buffer size 111
- set number of database latches 114
- set persistence of event notification
  - registrations 125
- set physical log buffer size 115
- set physical log volume size 91
- set roll forward on 222
- set server page buffer pages 113
- set short lock wait timeout 112
- set short locking off 101
- set short locking on 101
- set statistics collection off when using file 255
- set statistics collection off when using memory 255
- set statistics collection on and send to file 255
- set statistics collection on and send to memory 119, 255
- set statistics to derive 305
- set synchronization off 188
- set synchronization on 188
- set system volume extent size 89
- set system volume size 92
- set transactions allowed 116
- set up database directories and files 60
- setdbid 217
- signal\_block 79
- size limits of various configurable parameters 117
- software directories 282
- software root directory 285
- spin\_count 116
- ss.d 218
- start 218
- start database 219
- start roll forward archiving 231
- start roll foward archiving 222
- startdb 219
- stat 119
- statistics file 305
- statistics, derived statistics file 305
- statistics, get connection information 155, 255
- statistics, get database information 155
- statistics, get database list 154
- statistics, get defined statistics 255
- statistics, get environment information 203
- statistics, get lock information 155, 255
- statistics, get object information 155
- statistics, get process information 155
- statistics, get statistics from file 255
- statistics, get statistics from memory 255
- statistics, get storage volume information 155
- statistics, get thread information 155
- statistics, get transaction information 155, 255
- statistics, set collection off when using file 255
- statistics, set collection off when using memory 255
- statistics, set collection on and send to file 255
- statistics, set collection on and send to
  - memory 119, 255
- statistics, set statistics to derive 305
- statistics, view from memory 255
- statistics, view in file 255
- stop database 220
- stop roll forward archiving 232
- stop roll foward archiving 222
- stopdb 220
- swap\_threshold 82
- synchronize two databases 204
- sysinfo 304
- system 295
- system header files 290
- system information file 304, 324
- system utility errors, 70007999 424
- system volume 56, 295
- sysvol 92

## T

- trace\_comps 121
- trace\_entries 122
- trace\_file 122
- transaction 116
- transaction, set transactions allowed 116
- tuning parameters 106
- turn roll forward archiving on 239
- type, index 111

## U

- UNIX 92, 137
- UNIX communications parameter 343
- UNIX configuration parameter
  - considerations 320
- UNIX configuration parameter procedures 322
- UNIX configuration parameters for asserting values 329
- UNIX configuration parameters location 318
- UNIX configuration parameters to be set in special cases 325
- UNIX database server system parameter 350
- UNIX error logging and debugging
  - parameters 339
- UNIX mandatory configuration parameters 324
- UNIX statistics collection parameters 339
- UNIX user authentication parameter 343
- user 116
- user authentication 20
- user authentication process 32
- user authentication program structure 30
- user authentication sample programs, UNIX 35
- user authentication sample programs,
  - Windows 43
- user configuration file 322
- user privileges 35
- username 263
- using cluster administrator 565
- utility access privileges 18

## V

- vbackup 222
- vcopydb 245, 248
- verr 250
- verrindx 251
- VERSANT Manager errors, 40004999 401
- VERSANT mscs configuration guide 564
- VERSANT network errors, 30003999 397

- VERSANT Server errors, 10002999 388
- VERSANT\_auth 343, 364
- VERSANT\_be\_dbalogginglevel 105
- VERSANT\_be\_dbalogginglevel 105
- VERSANT\_be\_logfile 105
- VERSANT\_cfg 329
- VERSANT\_db 328, 354
- VERSANT\_db@ 337, 359
- VERSANT\_dbid 327, 354
- VERSANT\_dbid@ 336, 359
- VERSANT\_dbid\_node 326, 353
- VERSANT\_dbid\_node@ 335, 358
- VERSANT\_fe\_errlog 339, 360
- VERSANT\_rel 329, 357
- VERSANT\_rel@ 329, 357
- VERSANT\_root 325, 353
- VERSANT\_root@ 333, 357
- VERSANT\_service\_name 364
- VERSANT\_stat\_dbs 342, 363
- VERSANT\_stat\_file 339, 361
- VERSANT\_stat\_flush 342, 364
- VERSANT\_stat\_funcs 341, 362
- VERSANT\_stat\_stats 340, 362
- VERSANT\_stat\_time 341, 363
- VERSANT\_user 355
- VERSANTd 567
- version errors, 56005799 415
- view statistics from memory 255
- view statistics in file 255
- viewing the default groups 565
- vinstinf 251
- virtual attribute errors series 6600 423
- virtual system errors, 00010899 376
- vlicchk 252
- vlicvrfy 253
- vmovedb 253
- volume 117
- vstats 255
- vstream 269
- vtape 278

## W

- Windows configuration parameter
  - considerations 351
- Windows error logging and debugging
  - parameters 360
- Windows mandatory configuration
  - parameters 352

---

Windows optional configuration parameters 356  
Windows parameters 364  
Windows parameters for asserting values 357  
Windows parameters that must be set in special  
cases 352  
Windows statistics collection parameters 361  
Windows user authentication parameter 364

