Versant Vedding Usage Guide

Release 7.0.1.4

VERSANT

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**

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

Table of Contents

Vedding

This usage guide gives us an insight of “Vedding” which is a software module that performs synchronous database replication.

Following topics are covered:

- Concepts
- Procedures
- Usage
- Restrictions
- Recovery in case of Failure
- Network Partitioning Behavior
- Database Backup and Restore in a Replicated Database Environment
- Cautions

CONCEPTS

This section describes the functionality of "Versant Vedding,"(Fault Tolerant Server) a software module that supplements standard Versant functionality by performing synchronous database replication.

Synchronous database replication mirrors the contents of one database in another in a predictable and orderly manner. This provides either local or geographically remote redundancy, which protects against the unavailability of a database in the case of a system failure.

When you use synchronous database replication, the two databases kept in synchronization are called a "replica pair". If a crash occurs to one database, Versant will continue to use the other remaining database. When the crashed database comes back, Versant will automatically re synchronize the database with the database that did not crash and return to fully replicated operation. The resynchronization will occur while the database that did not crash continues to be updated.

Besides synchronous database replication, you can also protect yourself against system failure with roll forward archiving and/or asynchronous database replication.

For more information on roll forward archiving, please refer to Chapter "Versant Backup and Restore" in the *Versant Database Fundamentals Manual*.

For information about asynchronous database replication, please refer to the *Versant Asynchronous Replication Manual*.

PROCEDURES

Following are the steps required to perform synchronous database replication.

1. Create a database directory.
   You may want to create a new database directory for the database to be used for synchronous replication.
   - The database that you will use for replication cannot exist before you run the `createreplica` utility.
   - The synchronous replication database must be a group database.

   To maximize protection against system failure, you will probably want to locate the synchronous replication database on a machine different than the primary database.

   If you decide to use a new database directory for the synchronous replication database, use `makedb` normally to create the database directory and the database profile files. Per normal practice, after the profile files have been created, you can edit the server process profile file to specify non-default values for such parameters as system and log volume sizes.

   For example, to use `makedb` to create database directories and profile files for a group database named `replicadb`:

   ```
   makedb -g replicadb
   ```

For more information on makedb and server process profiles, please refer to the Chapter "Database Profiles" in the *Versant Database Administration Manual*.

2. Create a replica file.

   For each Versant installation that will use the primary database, create or edit a file that names the primary and synchronous replication databases. The replica file should be named `replica` and be placed in the Versant software root directory for that installation.

   You must create a replica file before creating a synchronous replication database.

   The replica file needs to be present in each machine that is involved in the replication. So if you have a client machine and two server machines, each with one of the pair of databases on it, all the three machines need to have a copy of the replica file.

   Once a replica file and a replica database have been created, Versant will automatically mirror the contents of the named databases each time a transaction commits.

   For each pair of primary and synchronous replication databases, the replica file should contain a line that names the databases. Use the following syntax for the primary and synchronous replication database, where node is the name of the machine containing the database:

   ```
   primarydb@node replicadb@node
   ```

For example, if the Versant software root directory is `/usr/local/versant`, the file `/usr/local/versant/replica` will contain the following entry for a database named `db1` located on a machine named `alpha` and its synchronous replication database `replicadb1` on a machine named beta:

```
db1@alpha replicadb1@beta
```

A connection made to either database in a replica pair will automatically cause a connection to be made to the other database in the pair.

The ordering of the names of the databases determines the order in which changes will be written.

Changes to the primary database will be written before changes are written to the synchronous replication database. By default, reads are done from the primary database. The database from which reads are made can be changed by calling `o_setpreferreddb()` or `setpreferreddb()`. If locks are asserted, they will be asserted in both databases.

The replica file can contain separate lines for any number of pairs of databases and synchronous replication databases.

You must use `database@node` syntax in the replica file to indicate the machines containing the databases to be mirrored.

In the replica file, any line starting with a # sign is considered to be a comment.

3. Create the synchronous replication database.

   Before beginning replication, you must use the `createreplica` utility to create the synchronous replication database and load it with the contents of a primary database.

For more information, on createreplica utility, please refer to the Chapter "Database Utilities" in the *Versant Database Administration Manual*.

4. Access either database in a normal manner.

   Once you have created a replica file and synchronous replication database, you can access either database using normal syntax and procedures.

   Each time you connect with either database of a replica pair with a begin session or connect database routine, Versant will look for the database name in the replica file and, if the name is found, automatically make a connection with the other database of the pair.

   When you perform an action on an object in either database of a replica pair, the action will be performed first on the object in the first database named in the replica pair in the replica file and then performed on the object in the second database. For example, if you lock an object in the second database, the lock will first be set in the database first named in the replica pair and then set in the second database.

5. Optionally set replica reporting.

When synchronous replication begins, by default a transaction will not report an error if a synchronous replication database is down. For example, if you try to make a connection to one of the databases in the replica pair and the connection fails because that database is down, the connection to the other database will succeed and access will be subject to standard operation during failure mode.

If you want to receive an error message if either of the replicated databases fail, then you can enable replica error reporting at the transaction level by doing the following:

1. Enable reporting for each database you want to monitor in the current database session.

   In C/Versant, enable reporting with the following function:

   ```
   o_err o_setreplicareporting( o_dbname db );
   ```

   In C++/Versant, enable reporting with the following method in PDOM:

   ```
   o_err setreplicareporting( o_dbname_const db );
   ```

2. Enable error reporting for a specific transaction by adding the option `O_REPORT_ERROR` to the commit option supplied to `o_xact()` or `xact()`.

   For example:

   ```
   o_xact( O_COMMIT | O_REPORT_ERROR, NULL);
   ```

   If a replica database being monitored is down, a commit will return error `4035: OM_TR_DBDOWN`.

At the next database session, the default behavior of no reporting will be restored until `o_setreplicareporting()` or `setreplicareporting()` is called again.

6. Administer both databases in a normal manner.

   Replica databases are normal databases. This means that you must manage both primary and synchronous replication databases as independent entities.

   For example, when you grant access privileges on a primary database, also grant privileges on the synchronous replication database.

   If the synchronous replication database runs out of space, use the `addvol` utility to add space as you would with any other database. If you run out of space in a primary database and add volumes to it, Versant will not check to see if the synchronous replication database needs space and will not automatically add space to it.

7. Optionally turn automatic synchronization off.

   By default, when either database in a replication pair goes down, the other starts a polling process. When the database that failed returns, the polling process will detect the return and resynchronize it.

   If you want to turn automatic synchronization off, you can use the `ftstool` utility.

   If you want to do manual synchronization, you can use the polling utility: `polling dbname`.

For more information, on ftstool and polling, please refer to the Chapter "Database Utilities" in the *Versant Database Administration Manual.*

8. Optionally compare the primary and synchronous replication databases. You can use the offline `comparedb` utility to compare the contents of a primary and synchronous replication database.

For more information on to comparedb, please refer to Chapter "Database Utilities" in the *Versant Database Administration Manual*.

9. Optionally remove a synchronous replication database.

   You can stop replication and remove a synchronous replication database with the `removereplica` utility.

For more information on to removereplica, please refer to the Chapter "Database Utilities" in the *Versant Database Administration Manual*.

USAGE

Create replica database

```
UNIX     createreplica (utility)
Windows  createrep (utility)
```

Delete replica database

```
UNIX     removereplica (utility)
Windows  removrep (utility)
```

Set replica reporting on for current session

```
c        o_setreplicareporting()
c++      setreplicareporting()
```

Set synchronization ON and OFF

```
UNIX     ftstool (utility)
Windows  ftstool (utility)
```

Set preferred database for read operations with synchronous database replication

```
c        o_setpreferreddb()
c++      setpreferreddb()
```

Restore defaults for read operations with synchronous database replication

```
c        o_unsetpreferreddb()
```

```
c++      unsetpreferreddb()
```

RESTRICTIONS

Following are restrictions on the use of synchronous database replication.

One of the replica databases must be a group database.

One of the replica pair must be a group database.

Only one replica can be created.

You can only create one replica of a database.

Cursor queries and event notification

Both cursor queries and event notification operate on a single database and, thus, are not fault tolerant.

Savepoints

You cannot set or undo a savepoint if synchronous database replication is in use.

Schema evolution requires that both databases be running

You cannot change schema when a database in a pair of replica databases is down.

Some routines can be used only on a single database

Some routines do not make sense for a pair of databases. For example, routines that manage serial numbers or user lists, that get information about a database, or that run a utility on a database can be used only on one database at a time.

Checkpoint commits and dropinst.

You cannot perform checkpoint commits or use the `dropinst` utility if synchronous database replication is in use.

RECOVERY IN CASE OF FAILURE

Following are recovery procedures in case one or both databases are not repairable.

If one database is down and is not repairable

1. Stop the database that is running:

   ```
   stopdb Updatabase
   ```

2. Remove the database that is not repairable:

   ```
   removedb -f DOWNdatabase
   ```

   - If the down database is the secondary database, then create a new replica database with the same name as the database that is not repairable:

     ```
     createreplica UPdb DOWNdb
     ```

   - If the down database is the primary database, then switch the order of the database names in the replica file (so that the secondary database is seen as the primary database.) Then create a new replica database with the same name as the database that is not repairable:

     ```
     createreplica UPdb DOWNdb
     ```

If both databases are down and are not repairable

1. Remove both databases:

   ```
   removedb -f db1
   removedb -f db2
   ```

2. Use the `vbackup` utility to restore the database previously backed up to a backup tape or file. (You do not have to backup both databases, just one or the other).

   ```
   vbackup -restore db1
   ```

3. Create the other database with `createreplica`:

   ```
   createreplica db1 db2
   ```

NETWORK PARTITIONING BEHAVIOR

Versant replication utilizes your network to manage distributed transactions across replicated databases. If your network fails, applications running local to a replicated database will conclude that the remote database is down and begin saving updates for later re-synchronization after the failure is corrected. This leads to a situation called *network partitioning* where objects may have conflicting updates in each database. If Versant detects that both databases in a replicated database pair have done ANY updates during the failure, then Versant will refuse to re-synch the databases.

You can avoid the network-partitioning problem through application design or by using the optional replication error reporting. This allows you to know in an application that a replica failure has occurred and then restrict updates to only one server. Three different update scenarios need to be considered:

1. If all applications execute on clients remote to replicated servers, then the network failure will probably cause all databases to be inaccessible and neither database will be updated. In this case, you don't need to use replica error reporting.
2. If applications doing updates run on only one server, then parallel object updates will not occur. In this case, you don't need to use replica error reporting.
3. If applications doing updates run on both servers, then a potential for parallel updates exists and error reporting must be used to prevent the applications on one server from updating during a replication failure.

   The transaction will not be committed if the database on which error reporting is set by call to `o_setreplicareporting` goes down.

DATABASE BACKUP AND RESTORE IN A REPLICATED DATABASE ENVIRONMENT

Vedding protects a database from single failures.

If one database in a replicated pair fails, it is typically restored automatically using automatic re-synchronization mechanisms. However, it is possible that a failure may continue for an extended period of time, after which it is better to remove the failed database with the `removereplica` command, and recreate it later using the `createreplica` command.

You may want to perform on-line backups and log archiving for a replicated database. In this case, you would only use the backup and log archives in a case where a failure to both databases occurs. If you choose to on-line backup and log archive only one database in the replicated pair, then a failure of the database being archived followed by a failure of the other database in the replicated pair will only allow a database restore to the latest log archive on the database where archiving was occurring.

The most conservative approach is to perform on-line backups and log archiving on both databases all the time. A more moderate approach is to only archive one database, and if the database being archived fails, immediately perform an on-line backup and begin log archiving on the remaining up database.

NOTE:- You should never attempt a database restore or log rollforward, using `vbackup` to one of the databases in an actively replicated pair.

CAUTIONS

- You must create a replica file and enter the primary and synchronous replication database names before initializing a synchronous replication database.
- Synchronous replication databases are normal databases. This means that you must manage both primary and synchronous replication databases as independent entities.

  For example, if either a primary or synchronous replication database runs out of space, use the `addvol` utility to add space as you would with any other database. If you run out of space in a primary database and add volumes to it, Versant will not check to see if the synchronous replication database needs space and will not automatically add space to it.
- You must use `database@node` syntax in the replica file to indicate the machines containing the databases to be mirrored.
- If you forget to include a pair of primary and synchronous replication database names in the replica file, no replication will occur and error E7220 will be generated by the createreplica utility.
- If you make a syntax error in the replica file, you will get an error message.
- If you make a consistency error in a replica file, you will not get an error message, but you will have problems later. All replication files in a network must be the same, or else several types of problems can occur.

  For example, if one replica file on one installation of Versant specifies a particular primary and synchronous replication database pair, but another installation does not, then changes made by some applications will be replicated but changes made by other applications will not be replicated.

  When you manage objects, actions are applied in the order in which a pair of databases are named in the replica file. For example, if you ask for a write lock on an object, it is first set on the object in the first database named in the primary/replica pair and then set on the object in the second database named in the primary/replica pair.

  If two applications use different replica files in which a particular database is designated as primary in one and as replica in the other, a deadlock can occur if both applications try to update the same object.
- Using `stopdb -f` is the same as a crash. Because replication is occurring, Versant will automatically attempt to restart the database stopped with `stopdb -f`.

  For example, suppose that `db1` and `db2` are a replica pair, both are running, and you stop `db2`.

  This will result in the following sequence:

  1. You stop `db2` with `stopdb -f db2`
  2. Versant immediately starts trying to restart and reconnect to `db2` (it will continue trying as long as `db1` is running).

3. As soon as Versant can restart and reconnect to `db2`, if automatic synchronization is on, it will resynchronize `db2` with `db1`.
4. The restarted `db2` becomes available again and resumes its role as part of the replica pair.

Index