Release Notes For Versant ReVind On Sun/Solaris

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

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

CHAPTER 1

Release Overview

This Chapter gives a brief overview of Versant ReVind 7.0.1 which is the latest version that works with Versant Database 7.0.1.

The Chapter covers the following:

- Enhancements
- Limitations
- Troubleshooting
- Upgrade Notes
- Restrictions and Suggestions

ENHANCEMENTS

Release 7.0.1

- Coherent with JDK 1.4.2. (JRE 1.4.2_07-b05).

Release 6.0.5

New License Manager integrated.

- Coherent with JDK 1.3.1 (JRE 1.3.1_02)

Release 6.0.1

- JDBC support is introduced.
- Java Stored Procedures (JSP) is supported.
- `NULL` predicate is supported.
- New `vsql.csh` and `vsql.sh` scripts to pickup the ODBMS frontend libraries instead of picking up the bundled frontend libraries.

Release 7.0.1.4

- Support for 64 bit Solaris platform
- Better Error Reporting. Errors display class name, view name, table name and also synonym name where ever applicable.

LIMITATIONS

Release 7.0.1

- O_U8B not supported.
- `IS [NOT] NULL` predicate only supported for OIDs.
- New `vsql.csh` and `vsql.sh` files are applicable only for 32 bit platforms. 64 bit platforms will still use the bundled frontend libraries.

TROUBLESHOOTING

To ensure an accurate startup of Versant ReVind 7.0.1 daemon, after installation, you should modify the file `/etc/services` to include the TCP/IP daemon name for Versant ReVind, that is if it does not already exist. These entries will be added to the /etc/services file if the Versant ReVind 7.0.1 installation was done by root.

For example:

```
  sqlnw           5020/tcp          #vsql listener
```

UPGRADE NOTES

If you are upgrading from previous version of Versant ReVind to 7.0.1, you need to perform following steps:

- To use databases of versions prior to 7.0.1, use the Versant Object Database 7.0.1 utility `convertdb` on the target database.

  Please refer to Versant Object Database 7.0.1 *Versant Database Administration Manual* for more information on using this utility.
- To check whether `convertdb` has succeeded, run `dblist` and ensure that the `dblist's` output, "db version" displays 7.0.1.3 for the converted database.
- Locate upgrade (in $TPEROOT/bin - upgrade for Solaris).

  Run this utility on the converted database.This will make the database usable with Versant ReVind 7.0.1.
- Verify that upgrade is completed successfully by running few ad-hoc queries.

RESTRICTIONS AND SUGGESTIONS

Multi-byte strings are interpreted as single byte strings

Versant ReVind interprets multi-byte strings as regular single byte strings. This implies that the first null encountered is considered as the end of the string.

Rollback after CREATE VIEW or DROP VIEW not supported

Versant ReVind supports views. The only restriction is that after issuing the command `CREATE VIEW` or `DROP VIEW`, the transaction should not be rolled back.
If an "undo" operation is desired, you can terminate that session of `isql` or `misql` without committing the transaction.
Per usual practice, to make a new view persistent or to drop a view, commit the transaction.
Although a view is dropped immediately after a `DROP VIEW` command, full cleanup of the dropped view from the base tables `systables` and `syscolumns` does not occur until a new `isql` or `misql` session is started.

For example:

```
ISQL> CREATE VIEW bookview (bookid, title)
         AS SELECT selfoid, title from book;
ISQL> commit work;
ISQL> DROP VIEW bookview;
ISQL> commit work;
```

Use the SYNONYM (instead of the real table) to create a VIEW

If you are creating a view and do not own the table, then you must create the view on the synonym for the table.
This is required, because the table names that are stored in the `sysviews` table are not double-quoted; if the view were to be created directly on the table, the case-insensitive table name would not be matched to the case-sensitive class names in a Versant database.
For example, the following is CORRECT:

```
ISQL> CREATE SYNONYM book FOR vsqldba."Book";
ISQL> CREATE VIEW bookview (bookid, title)
         AS SELECT selfoid, title FROM book;
ISQL> commit work;
```

The following is NOT CORRECT:

```
ISQl> CREATE VIEW bookview (bookid, title)
         AS SELECT selfoid, title FROM vsqldba."Book";
```

You cannot define a primary key in CREATE TABLE

When you create a table definition, you cannot define a primary key. The system adds an extra column to the set of columns defined by you in the table. This column is called `SELFOID` and is also the primary key. In the current release the foreign key will always be the primary key (`SELFOID`) of the referred table.
In this release the `CREATE TABLE` statement supports:

```
CREATE TABLE address
        ( street CHARACTER,
          city   CHARACTER);

CREATE TABLE address
        ( emp CHARACTER REFERENCES employee,
          street CHARACTER,
          city CHARACTER );

CREATE TABLE employee
        ( name CHARACTER DEFAULT USER,
          dob  TIME DEFAULT SYSDATE );

CREATE TABLE address
        ( street CHARACTER NOT NULL UNIQUE,
          city CHARACTER );

CREATE TABLE address (
        street, city )
        AS
        SELECT st, pin from customer;
```

In this release the `CREATE TABLE` statement does not support:

- Column level check constraints
- Table level constraints
- Table space
- Free percentage
- Storage Manager

For example:

```
Versant Action: IGNORE
CREATE TABLE address
        ( street CHARACTER COLLATE CASE_INSENSITIVE,
          city CHARACTER );

CREATE TABLE address
        ( street CHARACTER NOT NULL PRIMARY KEY,
          city CHARACTER );

CREATE TABLE supplier (
        supp_no INTEGER NOT NULL,
        name CHAR (30),
        status SMALLINT,
        city CHAR (20) CHECK (supplier.city <> 'MOSCOW')

CREATE TABLE student_courses (
        student_id INTEGER,
        teacher CHAR (20),
        course_title CHAR (30),
        FOREIGN KEY ( teacher, course_title)
                REFERENCES courses );

CREATE TABLE address (
        street CHARACTER,
        city CHARACTER )
        TABLE SPACE xxxx ;

CREATE TABLE address (
        street CHARACTER,
        city CHARACTER )
        PCTFREE 30;

CREATE TABLE address (
        street CHARACTER,
        city CHARACTER )
        STORAGE_MANAGER 'dfdf';

CREATE TABLE address (
        street CHARACTER,
        city CHARACTER )
        STORAGE_ATTRIBUTES 'fdsfj=xxxx';
```

NOTE:- Database should not have a `vsqldba` as a pre-existing user. NULL predicate only supported for OID columns.

Versant ReVind 7.0.1 works in FTS/Replica environment with following restrictions:

1. `schload` cannot be run if either of the databases in the replica pair is down. This is because `schload` involves some schema related changes introduced in the database. If any of the database is down, replication stops and hence `schload` fails.
2. `schload` returns `SM_NOT_IN_USERLIST` error when run for the first time on a replica pair. This error is returned by `SQLUTIL` when it tries to insert a dummy user (vsqldba) in both the databases. As this operation is not FTS compliant, `SQLUTIL` returns this error. In this case, after getting this error, please run `schload` again on the second database.
3. If the Versant ReVind databases are part of FTS/Replica setup and either of the database goes down while the query is being executed, the query does not complete. This is due to a known limitation in Versant cursors not being supported in FTS environment.

`schload` shows some warnings when run on a database

When `schload` is run on a database for the first time, it shows some warnings. Please ignore these warnings as these are harmless. The warnings will be removed in the next release of Versant ReVind.

Run `schload` on all the databases while connecting to multiple databases

When a dba tries to connect to multiple databases using ISQL/DHSERVER (Threaded) model, `dhserver` throws error `SM_NOT_IN_USERLIST`. This is due to a security restriction imposed by Versant Object Database 7.0.1.3. To avoid this error, run `schload` on all the databases participating in the multiple databases connection. This problem is observed only when the dba tries to connect to the database. In case of any other user, there is no need to do this.
Example: Say `db1` is the primary database. The owner and dba of this databse is `USER1`. Now if `USER1` tries to connect to `db1+db2+db3`, the `DHSERVER` throws error `SM_NOT_IN_USERLIST`. In this case either run `schload` on `db2` and `db3` or use some other user `(USER2)` to login to the databases.

CHAPTER 2

Platform Notes

This Chapter provides details of the system requirements and the directory structure for the Sun Solaris platform.

The Chapter covers the following in detail:

- System Requirements
- Directory Structure

SYSTEM REQUIREMENTS

Versant ReVind Release 7.0.1 for Sun/Solaris has the following system requirements:

Hardware Requirements

Platform	Sun 4 workstation.
Disk space	Minimum disk space for installation - 200 megabytes and For Temporary Files - 47 megabytes.
Memory	Runtime memory of 4 megabytes per user. Actual memory usage will depend on the size of the object cache used by the Versant ReVind(VSQL) process.

Software Requirements

Operating system	Solaris Version 2.9.
C++	SPARC Compiler C++ Version 5.6 Patch 117549-02 was used during development and testing of Versant ReVind.
Versant Object Database	An installation of Versant Object Database Release 7.0.1.

JDK Version 1.4.2 is required at the minimum. (Versant ReVind installation has JRE 1.5 bundled with it.)

DIRECTORY STRUCTURE

In order to install Versant ReVind 7.0.1 you have to run `install.bin` script which comes with the other files.

For example, if you choose `/usr/local/versant` as your installation directory, your Versant ReVind release directory will be:

```
/usr/local/versant/7.0
```

After installation, the environment variable `$TPEROOT` will point to the Versant ReVind release directory.

Under the Versant ReVind release directory, the installation program will create further directories and copy program files to them.

After installation, you cannot change the names or locations of the directories and files under the Versant ReVind release directory. However, you can move the entire structure if you reset `$TPEROOT` and perform again all of the configuration steps listed in the "Installation and Configuration" chapter.

If the installation is done by root then /etc/services file will be updated to add the entries for dhserver daemon service. If the installation is done by anybody other than root the /etc/services file will not be updated.

Following is an explanation of the directories and files that will be created under the Versant ReVind release directory, `$TPEROOT`:

Directories and Files Under $TPEROOT

$TPEROOT / vsql.csh

A system configuration script for users of C Shell. See the chapter "Installation and Configuration" for usage notes.

$TPEROOT / vsql.sh

A system configuration script for users of Bourne Shell. See the chapter "Installation and Configuration" for usage notes.

$TPEROOT / bin

The directory containing Versant ReVind binaries. These binaries include the following:

`isql`	Interactive Versant/SQL Tool (remote access.)
`dbload`	Copy data from a file into a database.
`dbdump`	Copy data from a database to a file.
`misql`	Interactive Versant/SQL Tool (local access.)
`schload`	Load schema and authorization information into a Versant database.
`sqlutil`	Allows a user to create persistent synonyms to provide case-insensitive references to tables for all user classes.
`sqlcrtidx`	Create indexes for extended tables. Used internally by `schload`.
`dhserver`	Versant ReVind daemon.
`upgrade`	Upgrades the previously configured databases.

$TPEROOT / bin / memdir

A directory for temporary files used by Versant ReVind during sorting operations. This directory must exist.

$TPEROOT/lib

The directory containing Versant ReVind libraries. These libraries include the following:

`libmap.so`	Versant ReVind shared library.
`libdata.so`	Versant ReVind shared library.
`libstubs.so`	Versant ReVind shared library.
`librdsm.so`	Versant ReVind shared library.
`librds.so`	Versant ReVind shared library.
`libsql.so`	Versant ReVind shared library.
`libsnd.so`	Versant ReVind shared library.
`libsnw.so`	Versant ReVind shared library.
`libdt.so`	Versant ReVind shared library.
`libpfe.so`	Versant ReVind shared library.
`liberr.so`	Versant ReVind shared library.
`libenv.so`	Versant ReVind shared library.
`libos.so`	Versant ReVind shared library.
`libdhgen.so`	Versant ReVind shared library.
`libss.so`	Versant ReVind shared library.
`libmm.so`	Versant ReVind shared library.
`libdhjava.so`	Versant ReVind shared library.
`libjsp.so`	Versant ReVind shared library.
`libstrm.so`	Versant ReVind shared library.
`libdhs.so`	Versant ReVind shared library.
`libjdbcnet.so`	Versant ReVind shared library.
`libdhserver.so`	Versant ReVind shared library.
`dherrors`	Versant ReVind error message file.
`dherrors_cust`	Versant ReVind error message file.
`sql conf`	You can use this to change the settings for the `ISQL/MISQL` sessions.
`vsql.jar`	Class files used by Versant ReVind, JDBC and JSP.

The directory also includes `sql_conf` that you can use to change the settings for the `ISQL/MISQL` sessions.

$TPEROOT / lib / schemas

`utility.sch`	Schema file for Versant ReVind utility classes.

`systab.sch,`
`systab1.sch`
	Schema files for base and extended tables loaded by schload.

$TPEROOT / lib / help_isql

This directory contains ascii help files used by the `isql` and `misql` utilities.

$TPEROOT / demo / addons / vsql

The directory containing the makecxx.com, demonstration programs and readme files. There are three demo subdirectories: `cxx, jsp and jdbc.`

$TPEROOT / demo / cxx

The `demo/cxx` directory contains the following C++/Versant demo files:

```
Makefile
README.txt
author.h
authsch.imp
authsch.sch
book.h
cindex*
cindex.cxx
input*
input.cxx
publish.h
sequel.h
```

$TPEROOT/demo/jsp

The `demo/jsp` directory contains the following Java Stored Procedures (JSP) demo files:

```
Makefile
README.txt
account.h
account.imp
calledsp.sql
callsp.sql
customer.h
demojsp.cxx
demojsp
dropsp.sql
sqlistmt.sql
transact.h
updatesp.sql
```

$TPEROOT/demo/jdbc

The `demo/jdbc` directory contains the following JDBC driver demo files:

```
README.txt
JDBCTest.java
JDBCTest.class
```

$TPEROOT/doc/addons

This directory contains Versant ReVind, Versant/ODBC, and Versant ReVind release notes in pdf format. To view these files, open them with a PDF viewer, such as Acrobat Reader.

CHAPTER 3

Installation and Configuration

This Chapter explains the installation and configuration of Versant ReVind 7.0.1 on Solaris machine.

The Chapter covers the following in details:

- If you are upgrading...
- Installation Procedures
- Configuring the Network
- Configuring Versant ReVind
- Installation Testing
- Installation Troubleshooting

IF YOU ARE UPGRADING...

If this is the first installation of Versant ReVind on this machine, then you can ignore this section.

If you are upgrading from a previous Versant ReVind release, then you need to do the following:

1. Stop the TCP/IP daemon.

   As root, stop the `dhserver` daemon with the following command:

   ```
   dhserver stop
   ```
2. Remove the previous release.

   As dba, remove the files and directories for the previous release.
3. Install the new release, except do not run `schload` on previously configured databases.

   As dba, install this Versant ReVind release using the procedures described in the following section, except do not run the `schload` utility on the Versant databases that have already been configured for Versant ReVind access.
4. Run the upgrade script on previously configured databases.

If you are upgrading from previous version of Versant ReVind to 7.0.1, you need to perform following steps:

1. To use databases of versions prior to 7.0.1, use the Versant 7.0.1 utility `convertdb` on the target database.

   Please refer to V*ersant Database Administration Manual* for more information on using this utility.
2. To check whether `convertdb` has succeeded, run `dblist` and ensure that the `dblist's` output, "db version" displays 7.0.1.3 for the converted database.
3. Locate `upgrade (in $TPEROOT/bin - upgrade` for Solaris)

   Run this utility on the converted database. This will make the database usable with Versant ReVind 7.0.1.
4. Verify that upgrade has completed successfully by running few ad-hoc queries.

INSTALLATION PROCEDURES

Versant ReVind incorporates a new licensing scheme from version 7.0.1 onwards. For information on obtaining the license file for Versant ReVind, go to `http://www.versant.com`.

Irrespective of whether you are upgrading the product or installing it for the first time, new licenses are needed.

For details about the procurement of licenses, please refer to `installSummary` file in root directory of Versant ReVind.

Versant ReVind Installation allows you to install the following components:

`dhserver`	Requires ReVind(VSQL component) license.
`isql`	Not separately licensed.
`dbdump`	Not separately licensed.
`dbload`	Not separately licensed.
`misql`	Requires ReVind(VSQL component) license.
`sqlutil`	Not separately licensed.
`sqlcrtidx`	Not separately licensed.

As dbsa — If you want to set up Versant ReVind 7.0.1

For UNIX installations, the database system administrator or "dbsa" is the user who as super user installed Versant on each machine in a network. There is only one database system administrator for a system of Versant installations.

The dbsa:

- Owns the `osc-dbid` file for a database system.
- 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.
- Owns all system information files for all installations in a network.
- Owns all database root directories for all installations in a network.

If necessary, change ownership of the files.

If you as dba are also the root user, then change the file ownership:

```
chown -R root vsql
```

Following are the steps required to configure Versant ReVind. If you are upgrading an existing installation, please follow all steps in the preceding section "If you are upgrading..".

Step 1> Modifying the appropriate Versant ReVind configuration script.

All the necessary settings will be done for you by the installer itself.

TPEROOT

The location of the Versant ReVind release directory.

For example, if you installed under `/usr/local/versant, TPEROOT` is:

```
/usr/local/versant/7.0
```

PATH

The location of Versant ReVind binaries, `$TPEROOT/bin`, will be added to your `PATH` variable.

LD_LIBRARY_PATH

The location of the Versant ReVind libraries, `$TPEROOT/lib`, `$TPEROOT/lib/odbms`, `$JDKHOME/jre/lib/sparc/server` (for 32 bit) and `$JDKHOME/jre/lib/sparcv9/server` (for 64 bit), will be added to your `LD_LIBRARY_PATH` variable.

TPEDATADIR

The location of a directory for temporary files to be used by Versant ReVind during sorting operations. Versant ReVind installation by default creates the directory `$TPEROOT/bin/memdir` for storing the Versant ReVind temporary files. However, you can use any directory for the temporary files, as long as it exists and the `TPEDATADIR` variable points to it.

There are separate scripts for C Shell and Bourne Shell.

C Shell — `vsql.csh`

Following is the configuration script `vsql.csh` for C Shell users:

```
setenv TPEROOT <installation_directory>

if ( $?TPEROOT ) then

    setenv PATH $TPEROOT/bin:$PATH

    setenv TPEDATADIR $TPEROOT/bin/memdir

    file `which oscp` | grep 64 > /dev/null
```

```
if ( $?VERSANT_ROOT && $status != 0 && `oscp –v` == ‘7.0.1’ ) then

    if ( $?LD_LIBRARY_PATH ) then

      setenv LD_LIBRARY_PATH $TPEROOT/lib:$LD_LIBRARY_PATH

    else

      setenv LD_LIBRARY_PATH $TPEROOT/lib

    endif

else

    if ( $?LD_LIBRARY_PATH ) then

      setenv LD_LIBRARY PATH $TPEROOT/lib:$TPEROOT/lib/odbms:

$LD_LIBRARY_PATH

else

      setenv LD_LIBRARY_PATH $TPEROOT/lib:$TPEROOT/lib/odbms

    endif

endif

    if ( $?CLASSPATH ) then

      setenv CLASSPATH $TPEROOT/lib/vsql.jar:$CLASSPATH

    else

      setenv CLASSPATH  $TPEROOT/lib/vsql.jar

    endif

    setenv THREADS_FLAG native

    unsetenv BIT

endif
```

Bourne Shell — `vsql.sh`

Following is the configuration script `vsql.sh` for Bourne Shell users.

```
TPEROOT=<installation_directory>

if [ "$TPEROOT" != "" ]

then

    PATH=$TPEROOT/bin:$PATH

    TPEDATADIR=$TPEROOT/bin/memdir

    file `which oscp` | grep 64 > /dev/null

    if [ "$VERSANT_ROOT" != "" -a $? -ne 0 –a `oscp –v` -eq “7.0.1” ]

    then

      if [ "$LD_LIBRARY_PATH" != "" ]

            then

                  LD_LIBRARY_PATH=$TPEROOT/lib:$LD_LIBRARY_PATH

       else

                  LD_LIBRARY_PATH=$TPEROOT/lib

      fi

    else

      if [ "$LD_LIBRARY_PATH" != "" ]

            then

                  LD_LIBRARY_PATH=$TPEROOT/lib:$TPEROOT/lib/
odbms:$LD_LIBRARY_PATH

      else
```

```
                  LD_LIBRARY_PATH=$TPEROOT/lib:$TPEROOT/lib/odbms

      fi

    fi

    if [ "$CLASSPATH" != "" ]

   then

      CLASSPATH=$TPEROOT/lib/vsql.jar:$CLASSPATH

    else

      CLASSPATH=$TPEROOT/lib/vsql.jar

    fi

    THREADS_FLAG=native

    export TPEROOT

    export TPEDATADIR

    export PATH

    export LD_LIBRARY_PATH

    export CLASSAPATH

    export THREADS_FLAG

    set BIT=

    export BIT

fi
```

Here the `$TPEROOT` is the parent directory of Versant ReVind 7.0.1 installation.

Usage notes for modifying the configuration scripts

- The configuration scripts are in standard UNIX.

- Note that the scripts do not specify the Versant database environment. Rather, they expect the Versant environment has already been set.

Step 2> As super user on each Versant ReVind server and client machine — Run script and configure network.

1. Run the appropriate configuration script.

   As super user, run either the C Shell configuration script `vsql.csh` or the Bourne Shell configuration script `vsql.sh`.

2. Configure the network.

   Perform the steps in the following section "Configuring the Network."

Step 3> As dba user — Run script and load key tables.

1. Run the appropriate configuration script.

   As dba user, run either the C Shell configuration script `vsql.csh` or the Bourne Shell configuration script `vsql.sh`.

2. Load key tables into at least one Versant database.

   You will need to load the Versant ReVind Mapper schema and authorization tables in the `$TPEROOT/lib/schemas` directory into a least one Versant database.

   To load the schema tables, use the Versant ReVind `schload` utility. Following is the general syntax for invoking `schload`:

   ```
   schload database_name
   ```

   Besides loading the schema tables, the `schload` utility will also grant query privileges to all database users. See the "Usage Notes" chapter for additional information on security.

3. As dba for all Versant databases, configure Versant ReVind.

   Perform the steps in the following section "Configuring Versant ReVind."

Step 4> As each user of Versant ReVind — Run the appropriate configuration script.

As noted above, before using Versant ReVind, each user must run either the C Shell configuration script `vsql.csh` or the Bourne Shell configuration script `vsql.sh`.

Be sure to run the appropriate configuration script before proceeding with the following configuration steps.

Step 5> As each user of Versant ReVind — Create table synonyms.

We recommend that each user create table synonyms, so that the user can refer to tables with a synonym rather using the fully qualified name in double quotes.

By default, tables must be accessed with the following syntax:

```
"tableowner"."tableName"
```

Elements are:

tableowner

All system tables are owned by the special user "vsqldba"; all utility and user tables are owned by dba.

tableName

Table names must be enclosed in double quotes.

Table names are case sensitive.

For example, if "jill" is the owner of table `Author,` access it with the following syntax:

```
SELECT * FROM "jill"."Author" ;
```

To avoid having to use this syntax, each user should create a synonym for all tables accessible with Versant ReVind by running the utility `sqlutil` as:

```
sqlutil -S db_1+db_2+...+db_n
```

In the above, substitute a list of database names containing Versant ReVind tables. Separate the names with the plus sign + and allow no spaces between elements of the list.

Afterwards, the user who ran `sqlutil` can refer to tables in the specified database simply by using the case insensitive table name without double quotes.

For example, to create a synonym for table `Author` in `db1`, run the following:

```
sqlutil -S db1
```

Afterwards, you can access table `Author` with any of the following syntax:

```
SELECT * FROM Author ;

SELECT * FROM author ;

SELECT * FROM aUthor ;
```

The `sqlutil` utility is located in the Versant ReVind bin directory. The Versant ReVind administrator can verify that the utility was successfully executed by querying the extended table: `syssynonyms`.

Step 6> As any Versant ReVind user — Optionally test the installation.

A sample demonstration object model and loading program are distributed with the Versant ReVind release. As an optional step, refer to the section on "InstallationTesting" for an explanation of how to use the sample application to test the installation.

CONFIGURING THE NETWORK

This section explains how to configure Versant ReVind to allow remote access.

Configuring the Versant ReVind network involves steps on the machines where Versant ReVind has been installed (the Versant ReVind "servers") and on the machines that will access Versant ReVind (the Versant ReVind "clients"). The same machine can be a Versant ReVind server as well as a Versant ReVind client.

Following steps are needed to configure your network:

As super user on each server machine

On each server machine, create a Versant ReVind entry in the `services` file.

As super user, on each machine on which Versant ReVind has been installed, you need to edit the network services file to associate a Versant ReVind service name with a port number.

Typically, the network services file is `/etc/services`.

The default service name for Versant ReVind is `sqlnw`.

The port number you specify should not conflict with any other network applications.

If you want to start multiple instances of the Versant ReVind daemon or want to override the default, you can specify a service name other than `sqlnw` and then specify the `-s` *name* option invoking the Versant ReVind daemon process. In any case, for each service name that you will use, you must create an entry for it in your `services` file and associate it with a port number.

For example, the following entry in a services file associates the default service name `sqlnw` with port number `5020`:

```
     sqlnw              5020/tcp
```

Applications that connect to Versant ReVind over the network must specify the same port number for the service name used in starting the `dhserver` process.

To enable access to Versant ReVind, start the Versant ReVind daemon.

The Versant ReVind `dhserver` daemon process must be running on a server machine if you want to access it from a remote machine or access it with a local component of Versant ReVind (such as the Versant/Interactive SQL query tool) that uses a network model of communication.

A single Versant ReVind daemon process can handle all connection requests from any number of Versant ReVind client applications.

You must start the daemon process as super user.

If you are using a Versant ODBMS Release other than 7.0.1, you must set the environment variable `VERSANT_REL` before starting the daemon.

The general syntax for managing the Versant ReVind daemon process is the following:

```
dhserver [option [option…] ] { start | stop | status }

option :: -e server_name | -s service_name | -q | -j service name | -h
host name
```

For example, if you have used the default service name of `sqlnw` in your services file, to start the `dhserver` process on a local machine, use the following command:

`dhserver start`

If you have used the default service name, to stop the daemon process:

`dhserver stop`

When invoking `dhserver`, you must specify one of the following command parameters:

`start`

Start the `dhserver` process.

`stop`

Stop the `dhserver` process.

`status`

Display the status of the `dhserver` process and any child process that it has spawned.

When invoking `dhserver` you can choose to fork a process or thread by specifying the environment variable `DH_SQL_THREAD` to be `"N"` or `"Y"`.

As super user on each client machine

On each client machine, create a Versant ReVind entry in the `services` file.

On each machine that will access Versant ReVind software, you need to edit the network services file to associate a service name with the same port number for the service name used in starting the `dhserver` process on the Versant ReVind server system.

Typically, the network services file is `/etc/services`.

The default service name for Versant ReVind is `sqlnw`.

The port number will have been specified in the `/etc/services` file on the server machine.

For example, the following entry in a services file associates the default service name `sqlnw` with port number `5020`:

```
sqlnw    5020/tcp
```

CONFIGURING VERSANT REVIND

After installing of Versant ReVind and configuring the network, you should configure Versant ReVind for each database that will be used. You must do this as the dba for each database involved.

There are two aspects to the configuration:

- Specifying the tables (classes) to be accessed using Versant ReVind.
- Granting SQL access privileges to Versant ReVind users.

As dba, specify classes to be accessed by Versant ReVind

By default, after the installation, Versant ReVind can be used to access all classes in any Versant database that is Release 7.0 or depending on your installation.

When Versant ReVind connects to a Versant database, the Versant ReVind Mapper queries the object schema from the database and maps it in a memory SQL representation. All user classes are mapped during the initialization of Versant ReVind.

When the set of classes that need to be accessed using Versant ReVind is a subset of the total set of user classes in the Versant database, the Versant dba can insert one entry in the table `VQNeededClasses` for every user class that needs to be accessed using Versant ReVind. Note that it is the class names that need to be inserted and not the table names. After all entries have been specified, the transaction should be committed.

For example, the following uses the Versant/Interactive Query Tool to specify that the classes `Author` and `Book` in database `db1` are to made accessible to Versant ReVind:

```
isql  db1

> insert into "VQNeededClasses" (className) values (‘Author’);

> insert into "VQNeededClasses" (className) values (‘Book’);

> commit work;

> select count(*) from "VQNeededClasses";

> COUNT(*)

> --------
>          2

> quit;
```

The next time Versant ReVind is invoked, the class `VQNeededClasses` is queried to determine whether a subset of the total user classes is to be visible using Versant ReVind. If there is even a single object of this class, the Versant ReVind Mapper will ensure that only that class is mapped.

As dba, grant SQL access privileges to Versant ReVind users.

For information on granting access privileges, see the section "Security" in the chapter "Usage Notes."

INSTALLATION TESTING

After installing and configuring Versant ReVind, the Versant ReVind administrator can perform some simple tests to understand the mapping of the Versant object model to the SQL model and verify the installation.

Under `$TPEROOT/demo` is a `README.txt` file that presents the object model that is used in the `Book-Author` demonstration program. This demo program allows you to load a user configurable number of `Book` and `Author` objects into a Versant database. The `README.txt` file lists some sample queries that can be executed using the Versant/Interactive Query Tool and the expected responses.

INSTALLATION TROUBLESHOOTING

This section attempts to list the typical problems that may be encountered during installation and configuration. The resolutions listed are the most likely of a set of possible resolutions.

Interpreting Error Numbers

Versant ReVind will raise errors from both Versant ReVind and the Versant database management system.

- Error numbers greater than zero are Versant ReVind errors.
- Error number less than `-310,000` indicate Versant errors, but to get the actual Versant error number, you need to multiply by `-1` and then subtract `310,000`:

  ```
  VERSANT error = ( -1 * error_number) - 310,000
  ```

  For example, if an association being traversed does not locate a persistent object, the Versant error 5006 ("Cannot find the object") is returned as Versant ReVind error `-315,006`. To obtain the actual Versant error:

  ```
  VERSANT error = (-1)(-315006) - 310000 = 5006
  ```

Cannot connect to a database

An attempt to connect to a Versant database using Versant ReVind fails.

Problem: A user tries to access a Versant database using Versant ReVind and encounters the error `SM_NOT_IN_USERLIST`.

Likely Resolution: Use Versant database utility `dbuser` to add the user to the database access list.

Startup error with misql or isql

The misql or isql command returns error -20000 or SQL_INTERNAL_ERROR.

Problem: A user tries to access a Versant database using Versant ReVind and encounters error -20000 or SQL_INTERNAL_ERROR.

Likely Resolution: Verify that the environment variable TPEDATADIR is set to a directory over which the user has write access (this directory is used by the memory storage manager of Versant ReVind to create a temporary data file while sorting large data sets).

For example, in csh:

```
setenv TPEDATADIR $TPEROOT/bin/memdir
```

Cannot access Versant ReVind over a network

- Network access to Versant ReVind returns a tcp bind error.

Problem: A user tries to access a Versant database using Versant ReVind (network access) and encounters an error 20212 in tcp bind 146.

Likely Resolution: Use the dhserver status command to ensure that the dhserver is running.

- The dhserver command returns an error when you try to start it.

Problem: When you try to start the dhserver using the command:

```
dhserver start
```

you receive the following error:

```
error: ld.so.1: dhserver: fatal: libsnr.so: can't open file: errno=2
```

Resolution: You get this error if you have inherited some other user's environment. To resolve this issue, add the $TPEROOT/lib directory to the LD_LIBRARY_PATH environment variable.

- The dhserver status command returns that the connection is refused.

Problem: A user invokes the `dhserver status` command to check the status of the Versant ReVind daemon `dhserver`, and the error returned indicates that the connection is refused.

Likely Resolution: Verify that the `dhserver` process is running using a UNIX command such as:

```
ps -ae | grep dhserver
```

If the `dhserver` process is not running, then start the `dhserver` process using the command:

```
dhserver start
```

Cannot start Versant ReVind

An attempt to use `misql` or `isql` utilities fails.

Problem: When you try to use `misql` or `isql`, you get the error message `"Terminating Error <E79>"`.

This error occurs because the Versant ReVind utilities `misql` and `isql` have been built with Versant Object Database Release 7.0.1 and you are trying to access a database other than a Release 7.0.1 database.

Resolution: Each user must set the environment variable `VERSANT_REL` to the appropriate release number. See the notes above for "Using a Versant database other than a Release 7.0.1 database."

Environment Variables

Make sure that the `LD_LIBRARY_PATH` environment variable is set.

If the `LD_LIBRARY_PATH` environment variable is not set to point to the `$TPEROOT/lib` directory, an error such as the one listed below can be expected:

```
isql mydatabase

ld.so.1: isql: fatal: libsql.so: can't open file: errno=2

Killed
```

All needed environment variables need to be set as specified in the *Installation and Release Notes* manual.

Index