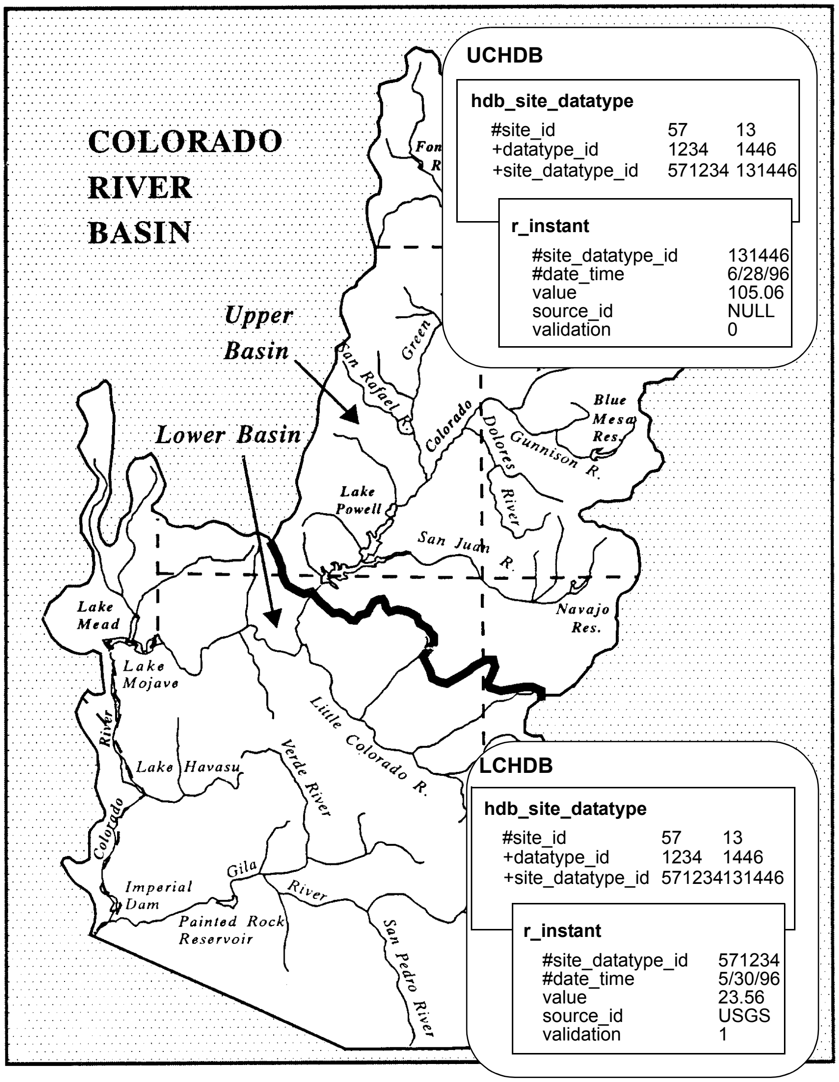
# Guidelines for Sizing, Placing and Creating New HDB Installations

[Next Page](#)

[Previous Page](#)



**UCHDB**

**hdb_site_datatype**

| | | |
|---|---|---|
| #site_id | 57 | 13 |
| +datatype_id | 1234 | 1446 |
| +site_datatype_id | 571234 | 131446 |

**r_instant**

| | |
|---|---|
| #site_datatype_id | 131446 |
| #date_time | 6/28/96 |
| value | 105.06 |
| source_id | NULL |
| validation | 0 |

**LCHDB**

**hdb_site_datatype**

| | | |
|---|---|---|
| #site_id | 57 | 13 |
| +datatype_id | 1234 | 1446 |
| +site_datatype_id | 571234 | 131446 |

**r_instant**

| | |
|---|---|
| #site_datatype_id | 571234 |
| #date_time | 5/30/96 |
| value | 23.56 |
| source_id | USGS |
| validation | 1 |

# Section 1 Introduction

Next Page

Previous Page

For every new installation of HDB, several decisions must be made to create an efficient, low-maintenance environment. Tablespaces must be sized properly to accommodate data, indexes, and rollback segments, and table extents must be large enough to accommodate initial loads of data. Improper use of these resources leads to inefficient, fragmented storage, and inconvenience to the Data Base Administrator (DBA). Tablespaces must be located on disk such that access to one tablespace does not compete with access to another. Similarly, important resources must be stored such that the loss of one resource (e.g., due to disk failure) does not put other resources at risk. Finally, resource allocation must be achieved within the system-layout structure mandated by the USBR's Oracle Standards and Guidelines document.

**This document briefly discusses:**

- Effective determination of data storage requirements.
- Effective and efficient placement of resources on disk.
- How to achieve these goals within the USBR structure.
- Creation of a new HDB installation.

**Note:** "RELEASE" in the remainder of this document refers to the top level directory in which the most recent HDB release was installed.

# Section 2 Sizing

## Estimating Time Series Table and Tablespace Sizes

Next Page

Previous Page

It is a good idea to have an estimate of how many rows will initially be loaded into each time series data table, and how many will accumulate over time. That information can be used to properly set parameters in the following files used to create HDB.

### createNEWHDB.sql

A sample version of this file can be found in RELEASE/oracle_script/SCHEMA/BASE_SCRIPTS/createNEWHDB.sql. It should be copied to $ORACLE_HOME/../../admin/<newDBname>/create/create<newDBname>.sql and edited as needed for your installation.

This file creates the hdb_data and hdb_idx tablespaces for HDB, which must be sized according to anticipated time series data load. Use the following sizing guidelines as a *rough* estimate.

*hdb_data tablespace*

300 bytes per row of time series data. Don't forget to include rows going into *r_base_archive* in your estimate.

*hdb_idx tablespace*

75 bytes per row of time series data. Don't forget to include rows going into *r_base_archive* in your estimate.

**Note:** These tablespaces are sized, in the released createNEWHDB.sql script, to hold about 5 million rows of time series data.

### timeSeries.ddl

This file holds the table creation statements for all time series data tables and any of their indexes. All tables are currently sized quite small, to hold only a few hundred rows in their first extent. The following should be considered in sizing the time series tables:

- Size the initial extent on both tables and date indexes (about 20 bytes per row) to hold the initial data load plus some anticipated future data. This will decrease fragmenting and improve performance.

- Tailor table sizing to each table based on anticipated load of data from different time steps, modeled vs. real, etc.

Next Page

Previous Page

# Other Tablespace Sizes

Other tablespaces which must be created are listed below. Their sizes have been determined by how they will be used. The creation statements and appropriate sizes are in the $ORACLE_HOME/dbs/create<dbname>.sql script.

- HDB_USER: The default tablespace for HDB users creating their own tables, which will happen rarely if at all; 5M.
- HDB_TEMP: The tablespace used by Oracle for sorting, etc.;200M.
- SYSTEM: The tablespace used by Oracle to store system information; 100M.

Next Page

Previous Page

# Section 3 Placement of Entities

Following is a list of guidelines for placing redo logs and tablespaces on disk. The first section reflects the USBR Standard Directory Structure for Oracle Databases. The second section is a less rigorous set of guidelines, but comprises a minimum to ensure data safety. If the database is on a RAID system, then data are redundantly stored and both disk contention and media failure are unlikely and separation of information onto different disks is less important. The directory structure and naming, however, should still be followed.

Next Page

Previous Page

## USBR Oracle Guidelines

The following information is paraphrased from the Reclamation IRM Documentation Standards and Guidelines, Oracle Standards and Guidelines updated on November 18, 1997. This section covers only the placement of entities directly related to the Oracle database. For information on installing the actual Oracle software, refer directly to the document.

### Redo Archives

The directory /redo_archive is for archived redo log files. Create a subdirectory for each database; e.g., /redo_archive/uchdb. Note that the archived redo log files should not be stored on the same disk as data, indexes and online redo logs. Specify the location of the archived redo logs in the init<dbname>.ora file, using the log_archive_dest parameter. **See"Initialization File" on page -77** for more information about the init<dbname>.ora file.

### System Tablespaces and System Directories

All nondata tablespaces and some database-specific system directories should be located under /ora/<database_name>; e.g., /ora/yakhdb. The directories under this tree are as follows:

### Online Redo Logs

The configuration of the online redo logs depends upon the nature and use of the database. At least two, and preferably three, groups of online redo log files should be created. Groups are identified by number; e.g., 1, 2.3. The members of each group are designated by letters of the alphabet; e.g., a, b, c, etc. Each group should have at least 3 members.

There are two possible configurations for online redo logs. For either configuration, you should create a directory /ora/<database_name>/<database_name>_redog, where g is the group number. For instance, /ora/yakhdb/yakhdb_redo1, /ora/yakhdb/yakhdb_redo2, etc.

Next Page

Previous Page

**Step 1.** Each redo log is located on a separate disk. All members of a group are on the same disk. This configuration:

- minimizes number of disks required,
- limits disk contention,
- provides recovery if a disk fails,
- when member *a* of a group is being archived, contention occurs with the other member (*b* or *c*) being written to on the same disk.

**Step 2.** Each redo log group member is located on a separate disk. That is, if you have three groups with three members each, nine disks are required. This configuration:

- eliminates disk contention,
- provides recovery if a disk fails,
- requires a lot of disks.

### Rollback Segments

Create a directory /ora/<database_name>/<database_name>_rbs for the rollback segments. For instance, /ora/yakhdb/yakhdb_rbs.

### User Tablespace

Create a directory /ora/<database_name>/<database_name>_users to hold the USER tablespace. For instance, the USER tablespace at Yakima might be: /ora/yakhdb/yakhdb_users/yakhdb_user.dbf.

### System Tablespace

Create a directory /ora/<database_name>/<database_name>_sys to hold the system tablespace. The system tablespace holds the data dictionary and other information.

### Tools

Any Oracle tools that are being used should be stored in /ora/<database_name>/<database_name>_tools.

### Control Files

In the init<dbname>db.ora file, you should specify three control files, each located on a separate disk. These control files are essential to the recovery of the database. They can be under any of the /ora directories (e.g. /ora/<database_name>_tools/ora/<database_name>/<database_name>_sys, and /ora/<database_name>/<database_name>_users) provided that these directories are mounted on separate disks.

Next Page

Previous Page

### Data Import/Export Directory

The directory /ora/<database_name>/<database_name>stg and its subdirectories should be used to hold files that are exported from a database or that contain data to be loaded into the database. For instance, /ora/yakhdb/yakhdbstg.

## Database Schema Tablespaces

Production data should be grouped by subject area or schema. A directory called /ora/<database_name>/<schema> can be added for each schema in the database. For most HDB installations, this would simply translate to; e.g., /ora/yakhdb/yakhdb.

### Data Tablespace

Store the HDB_DATA tablespace data files in a the directory /ora/<database_name>/<schema>/<schema>_data0n, where 0n is a sequential number. Unless your data files need to span more than one directory (or disk), you will probably just have one directory for the HDB_DATA tablespace; e.g., /ora/yakhdb/yakhdb/yakhdb_data01.

### Index Tablespace

The index tablespace (HDB_IDX) directory is named similarly to the data tablespace directory: /ora/<database_name>/<schema>/<schema>_idx0n; e.g., /ora/yakhdb/yakhdb/yakhdb_idx01. This directory should be on a different disk than the data tablespace directory.

### Temporary Tablespace

The temporary tablespace (HDB_TEMP) should be located in /ora/<database_name>/<schema>/<schema>_temp; e.g., /ora/yakhdb/yakhdb/yakhdb_temp.

# Minimum Requirements for Placing Database Entities

The following, minimum requirements should be met in order to ensure against data loss and to increase performance.

- A separate physical disk is needed for each of the following:
  - **HDB_DATA**,
  - **HDB_IDX**,
  - on-line redo logs (3 to 9 drives, depending on if you choose to put each redo member, or just each group, on a different disk).
  - archived redo logs (**log_archive_dest** in the database's init file)
- HDB_RBS can be on the same drive as HDB_DATA or HDB_IDX.
- HDB_TEMP can be on the same drive as HDB_DATA or HDB_IDX.
- SYSTEM can be on the same drive as HDB_DATA or HDB_IDX.
- HDB_USER can be on the same drive as HDB_DATA (or HDB_IDX, if simultaneous access of HDB_USER and HDB_IDX is unimportant).

Verify that the directory location of all tablespace data files in create<dbname>.sql is correct.

Next Page

Previous Page

# Section 4 Create Database File

In addition to any tablespace sizing edits already made to create<dbname>.sql, the following edits must be made:

- Set the init file name (the parfile parameter) to $ORACLE_HOME/../../admin/<dbname>/pfile/init<dbname>.ora, where <dbname> indicates the name of your database instance; e.g. yakhdb or hvrhdb.
- Change all occurrences of newhdb to <dbname>.
- Change all directories as appropriate to meet your needs, and, preferably, those guidelines laid out in **See"USBR Oracle Guidelines" on page -72**

Next Page

Previous Page

The file RELEASE/oracle_script/SCHEMA/BASE_SCRIPTS/initNEWHDB.ora is an example database initialization file that is appropriate for HDB installations. The file should be copied to $ORACLE_HOME/dbs/init<dbname>.ora, and edited as follows:

- Change all occurrences of newhdb to <dbname>, where <dbname> is the name of your database;
- Examine all directory paths and set them appropriately; see **See"Placement of Entities" on page -72** for guidelines on where to place the affected entities.

Next Page

Previous Page

Important! The "*compatible*" parameter in the sample *init.ora* file is set to 8.1.5. If you change this parameter to lower than 8.1.0, your snapshot installation (and refreshes from any associated master or snapshot) will not work. Snapshots are named differently when compatible is lower than 8.1.0 than when it is above 8.1.0; HDB2 code assumes that *compatible* is greater than 8.1.0.

You also need to create a link from the file in $ORACLE_HOME/dbs to $ORACLE_HOME/../../admin/<dbname>/pfile. CD to the $ORACLE_HOME/../../admin/<dbname>/pfile directory and do:

```
> ln $ORACLE_HOME/dbs/init<dbname>.ora .
```

# Section 6 Installing Oracle

This section is not intended to be a complete set of instructions for installing Oracle. It does, however, contain a few guidelines for ensuring that your Oracle installation and Unix environment is set up properly.

Next Page

Previous Page

**Step 1.** Read the instructions from the Oracle Unix Installation Guide thoroughly, and complete each step as specified. Any step that is missed or done improperly could cause the installation to fail without you knowing it.

**Step 2.** Ensure that your OS has a sufficient number of semaphores to handle the processes for all of your Oracle databases.

**a.** For each Oracle database that will be running (HDB being one), you need to determine the number of processes that can be running. This is determined by the "processes" parameter in the **init<dbname>.ora** file. (If you have only one database now, work with that. You can add semaphores at any time, when you need them for a new database.)

**b.** Examine the file /etc/system on your Oracle server.

- The value of **semsys:seminfo_semmsl** needs to be the number of processes from the database with the most processes, plus 10. So, if you have three Oracle databases, one with 200, one with 100, and one with 50 processes, you need to set **semsys:seminfo_semmsl** to 210.
- The value of **semsys:seminfo_semmns** needs to be: 2 x largest # of processes, plus sum of rest of database processes, plus 10 x number of databases. So, for the above scenario, **semsys:seminfo_semmns** needs to be: 2 x 200 + (100 + 50) + (10 x 3) = 580.

- If the semaphore values are not sufficient, you will have problems starting up the database instances. As root, edit /etc/system to reset the parameters, then reboot the machine to get them to take effect.
- You should check your OS guides to make sure you're not exceeding any system limits for these parameters.

**Step 3.** To install Oracle:

**a.** Do a Typical install of the Enterprise Server.

**b.** When this is complete, go back and do a Custom install of the Oracle X Client, and install the ProC compiler. This is an option under the Programmer package.

Next Page

Previous Page

# Section 7 Create the New HDB Installation

This section describes how to create a new HDB installation by running the associated scripts. It assumes that the scripts init<dbname>.ora, create<dbname>.sql, and timeSeries.ddl have all been modified according to the guidelines specified in this document.

Next Page

Previous Page

## As Unix user Oracle

**Step 1.** Set the environment variable ORACLE_SID to the name of your database instance. (This should be done in Oracle's .cshrc file. Make sure you source the file so that the variable gets set for the current login session.)

```
setenv ORACLE_SID <dbname>
```

**Step 2.** Go to the directory where your create<dbname>.sql script resides, $ORACLE_HOME/../../admin/<dbname>/create.

**Step 3.** Startup the instance using svrmgrl:

```
unix> svrmgrl

svrmgrl> connect internal
```

**svrmgrl> startup nomount pfile=../pfile/init<dbname>.ora**

**Step 4.** Run the create script from within svrmgrl:

**svrmgrl> @create<dbname>.sql**

**Step 5.** In another window, verify the output files generated by the create script: create<dbname>.log, and sql_scripts.log. You should see no errors in create<dbname>.log; in sql_scripts.log, you will see lots of errors regarding objects that do not exist; this is normal. If any problems exist, correct and rerun as needed.

Next Page

Previous Page

**Step 6.** Now startup the database fully from within your svrmgrl session:

**svrmgrl> startup pfile=../pfile/init<dbname>.ora**

**svrmgrl> quit**

**Step 7.** Change directory to $ORACLE_HOME/sqlplus/admin and run pupbld.sql as Oracle-user system (if your create script ran properly, the password is the name of the database):

**sqlplus system/<psswd>**

**sqlplus> spool pupbld.log**

**sqlplus> @pupbld.sql**

**sqlplus> quit**

**Verify the contents of pupbld.log, and then remove the file.**

**Step 8.** Start a listener on this Oracle database:

**lsnrctl start**

**Step 9.** Add an entry for this database, and any other remote databases to which you wish to connect, to your $ORACLE_HOME/network/admin/tnsnames.ora file.

**Step 10.** Test out the listener on your own database by doing:

**sqlplus system/<psswd>@<dbname>**

If you get into sqlplus successfully, the listener is responding.

**Step 11.** Set up your database server to shutdown the database for backups, bring it back up after backups, and automatically restart the database and listener upon reboot.

# As Unix user <DBA>

You should have a Unix login which corresponds to your HDB DBA. For instance, at Upper Colorado, UCHDB, the dba login is uchdba. If there is no such Unix user, create one. Under the home directory of the DBA, you should put the latest release of HDB creation scripts and application code. (CADSWES can help you with this step.)

Logged in as the dba, do the following:

Next Page

Previous Page

**Step 1.** Edit the .cshrc_hdb_app file in RELEASE/, and set two crucial environment variables:

**setenv ORACLE_SID <dbname>**

**setenv HDB_LOCAL <dbname>**

The HDB schema creation scripts will fail if these two variables are not set. Source this script from the DBA's .cshrc file to ensure the variables take effect at every login.

**Step 2.** Edit RELEASE/oracle_script/PERMISSIONS/local_user.sql. If you have identified the users on your HDB installation and know the roles that each should be granted, then modify this script accordingly. More specific directions on how to edit this file can be found in the document Hydrologic DataBase (HDB): Permissions and Roles. In lieu of reading this document, some brief guidelines for setting up new users follow:

Enter a **create** statement for each user you want in your installation. For each user create a **create user** command which consists of the following components:

**a.** Create the user:

Create user <user> identified by <password>

**b.** Set the default tablespaces:

default tablespace <HDB_user> quota unlimited on <HDB_user>

temporary tablespace HDB_temp quota unlimited on HDB_temp;

**c.** Grant connect privilege to each of the users:

grant connect to <user>;

**d.** Grant roles to users as follows:

- If user needs to edit r_base from **sqlplus** (not through an application), grant the role **savoir_faire**.
- If user needs to edit only monthly timeseries data from **sqlplus** (not through an application), grant the role **monthly**. (Grant either **monthly** or **savoir_faire**; not both)
- If user needs to run any DMIs (anything loading model data), grant the role **model_role**.
- If user needs to run any other data loading applications on HDB (e.g., hydromet data-loading), grant **app_role**.
- If user will change HDB_ and REF_ meta-data tables through the Meta Data Application, grant **hdb_meta_role**.
- If user will change only REF_ meta-data tables through the new Meta Data Application, grant **ref_meta_role**.

Next Page

Previous Page

**e.** Set user's default role(s). Default role(s) always include **connect**, and if user has been granted **savoir_faire** or **monthly**, then that role also. Default roles do not include **model_role**, **app_role**, **hdb_meta_role** or **ref_meta_role**.

**f.** An entire grant sequence for a user might look like this:

```
grant connect to caveman;
grant savoir_faire to caveman;
grant app_role to caveman;
grant hdb_meta_role to caveman;
grant model_role to caveman;
alter user caveman
default role connect, savoir_faire;
```

**Step 3.** Run the schema creation script, supplying the parameters prompted for:

**> RELEASE/oracle_script/create.script**

The parameters you need to supply are:

- Sys password: If your database creation script ran properly, this will be the name of your database
- DBA user name: e.g. uchdba

- DBA user password: supply the password you wish the Oracle DBA user to have. The user will be created with this password. (Remember what you type in!)

**Step 4.** In the same directory, verify the output from the creation script. Look at (*more*) each of the output files out1 -- out15. Also look at each of the *.out files; these are generated by all permissions-related sql scripts. When you get the message that the load of standard HDB data is complete, go to the STANDARD_DATA directory and check the log and out files. If anything appears to have not worked, determine which script generated the errors, fix the problem, and re-run the statement(s) which failed.

Next Page

Previous Page

At this point, you have a fully created, empty HDB. You now must run the scripts which set up the installation for the Meta Data Application. If you are an island site and don’t run these scripts, you cannot use the Meta Data Application. If you are a master or snapshot site, you *must* run these scripts to set up the proper links between distributed databases. *Do not load any data other than the automatically-loaded standard data into HDB until the Meta Data Installation scripts have been run*. See the Meta Data Application Documentation for instructions on using these scripts.

Once your installation is set up for the Meta Data Application, you may load additional meta data, and then time series data.