

HINTS for Rdb Controller Usage

On the CD

1. Installation instructions are on the inside of the CD cover.
2. Documentation for the tools is in the "/docs" directory of the CD in PDF Adobe format. Be sure you are selecting the documentation for your version of the tools.
3. The programs with instructions to assist in resolving user quotas issues (described below) are in the "/quotas" directory on the CD.

Startup/Installation

Most of the "startup" problems encountered by new users are in the initial setup and sizing of user quotas, system logicals, and work areas for the tools. What follows are some of the more common questions about which people have requested information.

Database Open

Be sure the database is "open" when running DBTune/DBAnalyzer/DBXact. This may be the normal state of affairs for your database if it is normally opened "manually", or if there are users constantly accessing the database. You will most likely see a problem if the database is open "automatic", and there is currently no one accessing it. This is indicated by a series of stat messages when the program is first launched, where the reported values are all "0". For example, the following would scroll past your screen or be placed in a log file:

```
*
*
$ run unloader
stat CS: 0
stat RC: 0
stat RD: 0
stat RG: 0
stat PS: 0
*
*
*
```

The solution is to open the database manually (even if it is currently automatic) for the duration of the program's run. This can be accomplished with the following command (where <db name> is your database logical or .rdb filename):

```
$rmu/open <db name>
```

Logicals

The following logicals are defined before running the tools. If the tool is running in batch mode, the logicals must be defined appropriately.

```
FREND$DBA$HOME-install directory for DBAnalyzer
FREND$DBA$SCRATCH-work area for DBAnalyzer
FREND$DBX$HOME-install directory for DBXact
FREND$DBX$DATABASE - database being worked on by DBXact
FREND$DBTUNE$HOME-install directory for DBTune
FREND$DBTUNE$SCRATCH-work area for DBTune
```

HINTS for Rdb Controller Usage

FREND\$DBTUNE\$PARAMS- "param" file for tuning parameters

FREND\$RDB\$IMPORT - database being worked on by DBTune

If you are running the "STANDARD" version of Rdb, be sure that your standard Rdb logicals are explicitly set by you BEFORE running the tools. This can be done by issuing the following manually, or in you login.com

```
\$@sys\$share:rdbvms\_setver S
```

If you are running the "MULTIVERSION" version of Rdb, you may have multiple versions of Rdb running simultaneously. The software will prompt you for the correct multiversion you wish to run the tool with, and set the Rdb logicals for you. For Rdb 7.0.x, simply enter "70", for Rdb 6.1.x, enter "61", etc etc. Do not enter "7.0" for example.

Quotas

Most users find when they first run the tools that the applicable user quotas have to be significantly increased. The manual recommends minimum values, and they are just that...minimum. There are two quota-monitoring programs in the "quotas" directory that may help you identify any quota problems.

Typically, PGFLQUO, BYTLM, ENQLM, and ASTLM are the quotas that are usually causing problems. FILLM can be a problem after tuning since one will typically have many more storage areas (i.e. rda and snp files) at this point.

If the tools exit abruptly due to quota issues when you first run them them, be sure to "clean out" FREND\$DBTUNE\$SCRATCH or FREND\$DBA\$SCRATCH respectively. These must be empty when the tools are starting up.

HINTS for Rdb Controller Usage

Param and mod_pad files

DBXact is useful, but not necessary for running DBTune. Sometimes this is not clear in the documentation. You can set the values manually in the param and mod_pad files. A default "param" file is in the DBTune installation directory, and you can copy it to another location and customize it for your use. The "param" file parameters are effectively for the "database as a whole", and are applied equally across all parts of the database (i.e. growth, snapshot percentage size, etc).

The mod_pad has the individual table/index parameters, along with the section for placing tables in the same area. This is also the parameter file for doing "placement via" associations, and the "do not place together" associations you want to impose (DBTune will automatically try to separate snp & rda files for tables, and their index files). A common method to getting an "initial" mod_pad file with all the tables and indexes in the correct format is to enter a directory/filename specification in the params file for a mod_pad file that does not yet exist. Run DBTune and let it generate an initial mod_pad for you, that you can then edit. Then re-run the tools "for real" with your "real" values.

(see DBTune section for recommendations regarding TUNE_FOR_COMPRESSION parameter)

DBDISKS

The params file is where you (in the DBDISK section) can tell DBTune to use only x blocks of space, what types of Rdb files should (or should not) go on a disk, etc. This section has been a little confusing for new users, so I will go over it in more detail.

DBTune assumes the fastest device is named first (DBDISK01), the second fastest is named second (DBDISK02), etc. Keep this in mind. Usually it is a good idea to place the disk/directory for your current "root" file as DBDISK01, but using various qualifiers described below, one can pace it on any disk.

DBDISKs are usually separate disk:[dir] specifications, but can also refer to different directories on the SAME DISK. For example

```
DBDISK01=DISK1:[RDB.RDA]
DBDISK02=DISK1:[RDB.SNP]
```

To force different type of files into specific directories, you must explicitly modify EACH DBDISK line. Remember that a line w/out additional qualifiers means that ANYTHING can be placed at that location...and the tool will try to use such an entry since it is attempting to spread out the database I/O.

For example, if you want to place SNP and RDA files in unique directories on EACH disk, you need to do something like the following for EACH DBDISK entry

```
DBDISK01=disk1:[rdb.system]/SYSRDB/
DBDISK02=disk2:[rdb.rda]/SYSRDA/TBLRDA/IDXRDA/500000/
DBDISK03=disk2:[rdb.snp]/SYSSNP/TBLSNP/IDXSNP/250000/
DBDISK04=disk3:[rdb.rda]/SYSRDA/TBLRDA/IDXRDA/500000/
DBDISK05=disk3:[rdb.snp]/SYSSNP/TBLSNP/IDXSNP/250000/
Etc
```

Here is a further description of what the above entries are trying to specify:

DBDISK01= ONLY the root file goes here. It allows no other file type. The tool uses any available free space (since no space qualifier is present).

DBDISK02= in the [rda] directory of disk1, allow system, table, and index RDA files ONLY. You have 500,000

HINTS for Rdb Controller Usage

blocks available for these files (or so you say). Note that since DBDISK02 and DBDISK03 refer to the same disk, you have to "split" the actual available disk space between them. In this example, the actual usable free space for disk1 would appear to be $500,000 + 250,000 = 750,000$ blocks total.

Note the DBTune takes the very conservative view of assuming we are talking about FREE SPACE CURRENTLY ON THE DISK. In fact, if you know there are 2M blocks currently used, and that you are going to REUSE this entire space (by dropping whatever takes up the 2M blocks, freeing up this space, are recreating it during the tune), you can add this "reusable" space to the available free space figure ($2,000,000 + 750,000 = 2.75$ M blocks really available for use/reuse on disk1). The operator must now decide how to split it up between the different DBDISK entries for this physical disk as previously described.

You will get a warning message during the script "generation" portion about there being insufficient free space (since in the script generation phase the object has not yet been removed), but you can ignore it.

DBDISK03=in the [.snp] directory of disk1, allow system, table, and index SNP files ONLY. Ditto description on the available space the product can use.

DBDISK04=in the [.rda] directory of disk2, allow system, table, and index RDA files ONLY. Ditto on space...

DBDISK05=in the [.snp] directory of disk2, allow system, table, and index SNP files ONLY. Ditto on space....

Etc

Etc

Note: In the next version of the, DBTune (V6) will consider the free/reusable disk space automatically.

Program memory work areas

If you get an error message referring to "inform\$name.pages" (or any of the inform\$XXXX messages) being insufficient when you start (the tune program in particular may show this since it uses the most work space), then one of the default internal "work areas" is too small. This can be due to the size or complexity of your database. You can increase the value in the DBTUNE $_{exx}$.DCL, clean out the FRENDS\$DBTUNE\$SCRATCH directory, and run the script generator program again. There are several examples of values typically reassigned in that file already. Contact us and we can guide you in the proper choice of a value.

DBXact scanning

The default interval for DBXact monitoring is 10 seconds. This is great for demos, poor for real world use. A value more like 60s is more realistic. Set scan duration to 6-12 hours during typical "active periods". Pick something representative, and not an "unusual" activity like a batch program that only runs 1X/month. Short monitoring of a few hours is often better than running it for 24-48-72 hours especially if very little is happening during most of this period!

DBTune tuning

You can run DBTune in batch mode using the DBTune_batch.com file in FRENDS\$DBTUNE\$HOME. A similar file for DBAnalyzer also exists.

It is best to start with the SQL tuning method. It does the entire database including RDB\$SYSTEM. It can adjust all the basic parameters of the database. When tuning more than a few database objects, we find that SQL mode is usually the fastest method as well

HINTS for Rdb Controller Usage

RMU tuning is for tuning small parts of the database. You have to take more care in setting your tuning parameters and other logicals (like computed by fields, segmented strings, etc). Logicals that must be set for these features are described in the manual.

Compression should be ENABLED in the params file when you first start testing via the TUNE_FOR_COMPRESSION parameter. By default this parameter is set to "N" in DBTune. However, compression is ON by default in Rdb. While turning off compression can give you a slight performance improvement in the final tuned database, it can result in MUCH LARGER storage areas (which often surprises/startles new users) since they will be sized for UNCOMPRESSED record lengths. Enabling compression gives you the Rdb default, which you probably have now.