Using IBM Informix on a Workstation with DBXten

Table of Contents

Executive Summary

Having benchmarked our DBXten DataBlade on a powerful and expensive IBM System x computer (documented in a previous 2010 report[1]), we wanted to explore what performance advantages could be offered by this BCS database plug-in on a commonly available and inexpensive office PC. Our objective was to assess the loading and retrieval performance of IBM Informix Dynamic Server (including the compression features introduced by IBM in 2008), *with and without* DBXten, on a simple workstation.

These tests on the PC showed that:

1. Using the DBXten DataBlade required 15 times less storage space than "native Informix" (i.e., Informix without DBXten).
2. The DBXten DataBlade reduced load times of indexed tables to less than 40% of the load times taken by native Informix.
3. The automatic data compression in DBXten actually improved multi-dimensional query performance *by almost an order of magnitude.*

Since databases are usually multi-user intensive platforms, we also examined DBXten's performance in concurrent query testing ranging from 2 to 30 simulated users. We found that as we gradually increased the number of users up to 30, the performance gains *increased* using DBXten, and were always *more than an order of magnitude!* Truly impressive results considering that we used a $600 PC. Keep in mind that, as in our previous report, the only change between the comparative tests was addition of the DBXten DataBlade. *No* additional customization was necessary to achieve these results.

The DBXten DataBlade provides compact storage, fast loads, and quick queries for large datasets on a wide range of computing platforms, from inexpensive desktop PC's up to and including large and costly multi-server environments.

Indeed, comparing the results in this report with those obtained earlier using the IBM System x computer, we see that Informix *with* DBXten on a readily available desktop PC can offer performance that is comparable to that obtained on a far more expensive configuration.

[1] See http://www.barrodale.com/bcs/whitePapers/Impact_of_DBXten_on_IBM_Informix_Performance.pdf

Introduction

In this report we compare the load and query performance of IBM Informix with and without the Barrodale Computing Services (BCS) DBXten DataBlade[2]. Computations with a scientific dataset occupying up to 440 million database rows were run on a BCS workstation (an AMD Athlon 64 dual core). This benchmarking was conducted after we had completed similar tests[3] accessing an IBM System x configuration located at the IBM Innovation Center in San Mateo, CA.

Of primary interest to us was how the performance of DBXten would scale as the number of rows in the database increased. For each of eight increasing table sizes ranging from 55 million to 440 million rows, tables were loaded and the same set of five queries was executed, firstly by a single user and then by (simulated) multiple concurrent users. The Informix load and query performances, with and without DBXten, were then compared. In short, DBXten scaled very well on the office PC at BCS. Loading with DBXten required much less space (by a factor of 15 — largely because of a factor of more than 200 in index space reduction) and load times were faster by a factor of 2.5. Individual query times with DBXten were faster by factors ranging from 2.6 to as much as 32, with an average factor of 7.9. The DBXten cumulative query times for each of the eight tables were more than an order of magnitude faster than for native Informix. Finally, performance trials with up to thirty (simulated) concurrent users were run on the 220 million row table. These trials demonstrated that enhancing Informix with DBXten provided answers to queries more than 17 times faster when two concurrent users were querying on the BCS workstation, and more than 20 times faster for three to thirty users.

Surprisingly, our tests also showed that DBXten on the PC provided some actual *timing gains* over those reported previously on the large and expensive hardware at San Mateo. Clearly, running Informix *with* DBXten on an inexpensive computer provides a very efficient alternative whenever access to a conventional fault tolerant server is unavailable.

The body of this report contains technical specifications of the comparison testing and the computational results obtained, and the three Appendices provide further details on the Informix configuration parameters at BCS, sample data file header information, and the querying source code for when the DBXten C API was used. For readers of this report who wish to focus mainly on the outcomes of our testing, we direct attention to the Tables and Figures on pages 8–14.

[2] Update: In December 2011 BCS was awarded US Patent 8077059 for DBXten.
[3] The tests on the more powerful IBM configuration occupied up to 1,447 million database rows.

Computing Environment

Server

CPUs: 1 AMD Athlon 64 X2 3600+ Dual-Core, 2.0 GHz, 2x256KB L2Cache, 64 bit
RAM: 2GB DDR2 667MHz
OS: Linux kernel 2.6.18 (Centos 5)
Disk: 1x160GB Seagate Barracuda 7200 rpm 3GB/s SATA internal hard drive with cooked filesystem; 1x1TB Western Digital 7200rpm SATA internal hard drive, accessed by Informix as a raw disk

Informix

Informix Dynamic Server version 11.50FC6 was used throughout, and the `onconfig` files used are presented in Appendix A.

The `onconfig` file used was the standard version distributed with Informix, with the following parameter changed:

```
PHYSFILE 500000
```

Number of `BUFFERS` in `BUFFERPOOL` statements was set to `100000` and the logical log size was set to `10000 kb`.

The Informix dbspaces and sbspaces were defined as follows:

Name	Type	Size (kb)	Offset (kb)	Location
rootdbs	dbspace	200,000	4	raw partition 1
temp1db	dbspace	2,000,000	200,004	raw partition 1
temp1sb	dbspace	200,000	2,200,004	raw partition 1
temp1sbifmx	dbspace	150,000	2,400,004	raw partition 1
miscdbtab (1)	dbspace	100,000,000	2,550,004	raw partition 1
physlogdb	dbspace	1,000,000	102,550,004	raw partition 1
loglogdb	dbspace	2,000,000	103,550,004	raw partition 1
miscsbtab	sbspace	20,000,000	105,550,004	raw partition 1

Table 1: Informix DbSpace and Sbspace Definitions

The BCS machine had two disks, and all dbspaces / sbspaces were built in a single raw partition on the second disk.

The Data

The data was generated using a high resolution model developed by the Ocean Circulation and Climate Advanced Modelling Project, using the OCCAM data selector at http://www.noc.soton.ac.uk/JRD/OCCAM/EMODS/select.php. The parameters shown in the image overleaf were used to direct the generation of data:

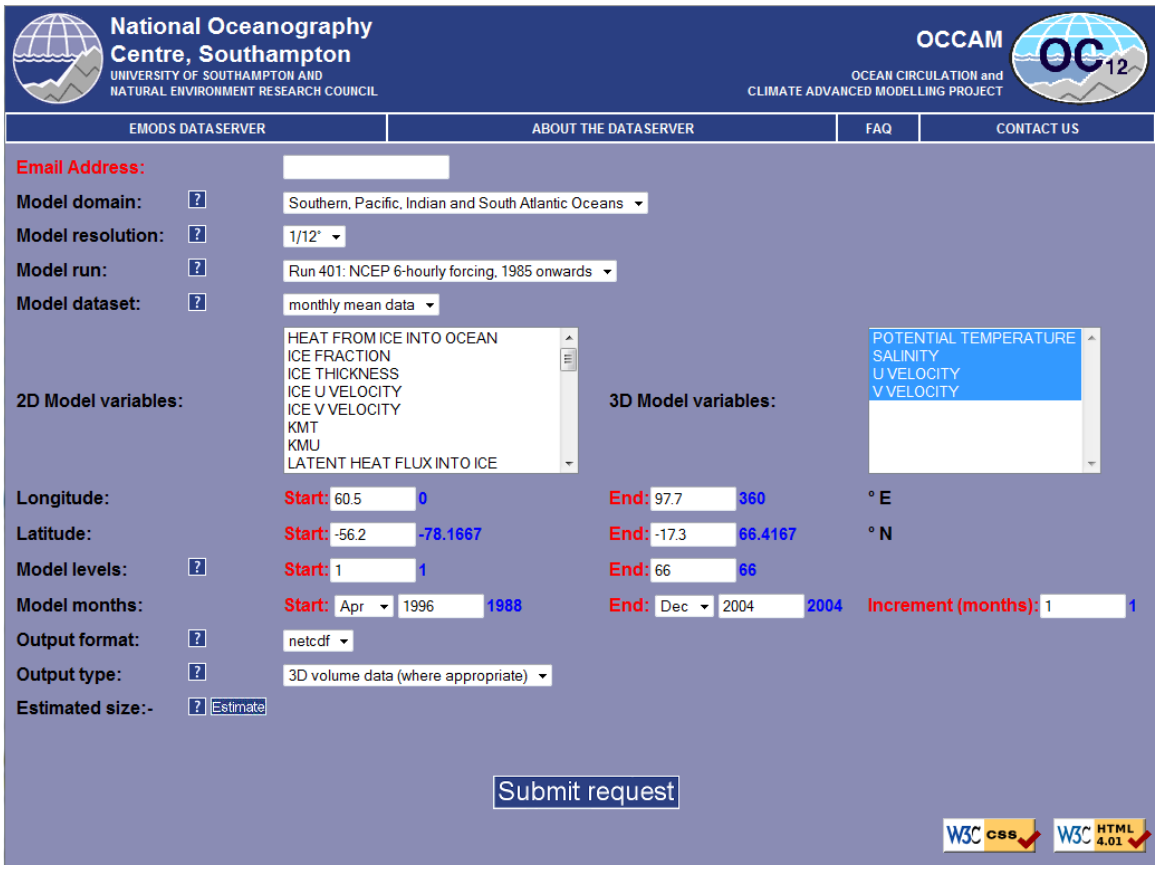

Figure 1: OCCAM Data Selector, with values filled in

This request[4] resulted in the generation of 105 netCDF files, one for each month between April 1996 and December 2004. The header information from one of these netCDF files is provided in Appendix B. Each of the netCDF files was then converted to a comma-separated (CSV) text file, with each line containing values from the six variables:

- TIMESTEP
- DEPTH
- LATITUDE_T
- LONGITUDE_T
- POTENTIAL_TEMPERATURE__MEAN_
- SALINITY__MEAN_

Native Informix Table

The native Informix table had the schema shown in Table 2 on the next page:

[4] The request was actually submitted as three individual requests, covering different periods of time.

Column	Type
timeval	integer
depth	float
latitude	float
longitude	float
temperature	float
salinity	float

Table 2: Native Informix Table Schema

The tables were defined using the SQL:

```
CREATE TABLE "barrodale".occam_conventional_hpl_ind_comp_$N
  (
    timeval integer,
    depth float,
    latitude float,
    longitude float,
    temperature float,
    salinity float
  )
  IN miscdbtab
  EXTENT SIZE $SIZE NEXT SIZE $SIZE LOCK MODE PAGE;
```

The values $N and $SIZE are given in Table 3:

Actual number of rows	$N	$SIZE
55,109,736	50,000,000	1,000,000
82,664,604	75,000,000	1,500,000
110,219,472	100,000,000	2,000,000
137,774,340	125,000,000	2,500,000
165,329,208	150,000,000	3,000,000
192,884,076	175,000,000	3,500,000
220,438,944	200,000,000	4,000,000
440,877,888	400,000,000	8,000,000

Table 3: Values of $N and $SIZE

Native Informix B-Tree Indexes

For each Informix table four B-Tree indexes were built, one on each of the spatio-temporal columns timeval, latitude, longitude, and depth. The following SQL was used to define (attached) indexes on the tables:

```
CREATE INDEX
"barrodale".occam_conventional_hpl_ind_comp_$SIZE_depth_idx
    ON "barrodale".occam_conventional_hpl_ind_comp_$SIZE
    (depth) FILLFACTOR 50;

CREATE INDEX "barrodale".occam_conventional_hpl_ind_comp_$SIZE_dt_idx
```

```
    ON "barrodale".occam_conventional_hpl_ind_comp_$SIZE
    (timeval) FILLFACTOR 50;

CREATE INDEX
"barrodale".occam_conventional_hpl_ind_comp_$SIZE_latitude_idx
    ON "barrodale".occam_conventional_hpl_ind_comp_$SIZE
    (latitude) FILLFACTOR 50;

CREATE INDEX
"barrodale".occam_conventional_hpl_ind_comp_$SIZE_longitude_idx
    ON "barrodale".occam_conventional_hpl_ind_comp_$SIZE
    (longitude) FILLFACTOR 50;
```

DBXten Table

For the DBXten trials the table had a single column, a so-called "DSChip" type column with the following schema definition:

Column	Type	Precision
timeval	integer	
depth	float	.01
latitude	float	.001
longitude	float	.001
temperature	float	.01
salinity	float	.000001

Table 4: DBXten DSChip Schema

One of the features of DBXten is that it is able to exploit any lack of precision in a particular dataset. The precision values shown above were chosen to correspond with the actual precision of the data in the input dataset (as converted from the original netCDF files).

The SQL used to define the table was:

```
CREATE TABLE occam_chips_${SIZE}_0_TM(occamchip dschip) IN miscdbtab
PUT occamchip IN miscsbtab;
```

DBXten Index

The SQL below was used to define an R-Tree index on the DSChip's:

```
CREATE FUNCTION occam_cube(chipIn DSChip)
      RETURNING dsbox WITH (NOT VARIANT)
   RETURN
DSAsBoxString(chipIn,'timeval,latitude,longitude,depth')::dsbox;
END FUNCTION;
CREATE INDEX occam_chips_${SIZE}_0_TM_idx ON
      occam_chips_${SIZE}_0_TM(occam_cube(occamchip) dsbox_ops)
      USING rtree;
```

Loading Trials

Native Informix Loading Trials

Table 5 defines the loading trials that were performed on native Informix tables:

Case Name	Rows (N)	netCDF Files	Indexes	Pre-indexed?
55M_B_R	55,109,736	4	4 B-Tree	yes
83M_B_R	82,664,604	6	4 B-Tree	yes
110M_B_ R	110,219,472	8	4 B-Tree	yes
138M_B_R	137,774,340	10	4 B-Tree	yes
165M_B_R	165,329,208	12	4 B-Tree	yes
193M_B_R	192,884,076	14	4 B-Tree	yes
220M_B_ R	220,438,944	16	4 B-Tree	yes
440M_B_ R	440,877,888	32	4 B-Tree	yes

Table 5: Loading Trials – Native Informix

Each of these loads was performed using the High-Performance Loader in "Deluxe Mode". Although the High-Performance Loader offers a much faster "Express Mode", which disables indexes during loads, this option was not used since it seemed inappropriate for an environment (such as a sea-floor observatory) where data is being ingested continuously and where there is a continuous need to be able to query the data.

Informix row compression was used throughout. Specifically, the procedure was:

1) Load 3,000,000 rows into the previously empty, non-compressed table.
2) Turn on compression using:
   ```
   execute function sysadmin:task("table compress repack shrink",tablename)
   ```
3) Load the remaining rows.

This procedure ensured that a dictionary of adequate size was built to allow the remaining rows to be effectively compressed.

DBXten Loading Trials

Table 6 on the next page defines the loading trials that were performed on the DBXten tables; the case names reflect the name of the comparable native Informix loading trial:

Case Name	Rows (N)	netCDF Files	Indexes	Pre-indexed?
55M_B_R	55,109,736	4	1 R-Tree	yes
83M_B_R	82,664,604	6	1 R-Tree	yes
110M_B_ R	110,219,472	8	1 R-Tree	yes
138M_B_R	137,774,340	10	1 R-Tree	yes
165M_B_R	165,329,208	12	1 R-Tree	yes
193M_B_R	192,884,076	14	1 R-Tree	yes
220M_B_ R	220,438,944	16	1 R-Tree	yes
440M_B_R	440,877,888	32	1 R-Tree	yes

Table 6: Loading Trials – DBXten

The DBXten tables were loaded with a CSV file loading client program, which is distributed with the DBXten product.

Note that each of the DBXten trials in the tables above have corresponding native Informix loading trials to which load times and space consumption can be compared. These comparisons were performed, and the results are presented in the next two sections.

Loading Time and Space Comparison

The table below presents the load times and table and index space consumption for each of the trials:

	Total Space (2K pages)			Index Space (2K pages)			Load Times (hh:mm:ss)		
Case	Native	DBXten	Ratio	Native	DBXten	Ratio	Native	DBXten	Ratio
55M_B_R	1,542,593	102,044	0.0662	562,147	2,633	0.004684	0:28:54	0:10:53	0.37639
83M_B_R	2,330,727	152,987	0.0656	837,420	3,945	0.004711	0:42:54	0:16:13	0.37795
110M_B_ R	3,119,120	203,865	0.0654	1,113,120	5,250	0.004716	0:56:26	0:21:40	0.38384
138M_B_R	3,906,106	254,802	0.0652	1,388,402	6,563	0.004727	1:09:47	0:26:49	0.38423
165M_B_R	4,693,232	305,456	0.0651	1,663,791	7,869	0.004730	1:23:30	0:32:34	0.38999
193M_B_R	5,480,552	356,177	0.0650	1,939,113	9,196	0.004742	1:37:18	0:37:35	0.38622
220M_B_ R	6,267,280	407,140	0.0650	2,214,515	10,510	0.004746	1:50:46	0:43:20	0.39117
440M_B_R	12,557,789	815,447	0.0649	4,417,477	21,036	0.004762	3:39:19	1:27:16	0.39791

Table 7: Load Time and Space Comparison

Ratios were quite consistent, with total space ratios ranging from .065 to .066. Load time ratios were consistent, ranging from .360 to .390.

Query Trials

Query performance, with and without DBXten, was assessed by running a variety of queries against each of the tables loaded as described in the previous sections.

For each of the cases listed in Table 7, five queries were issued in succession against the native Informix tables and then against the DBXten tables. In between each query invocation the Informix engine was stopped and restarted as follows:

```
onmode -ky
oninit
onmode -e flush
onmode -F
```

and file caches were cleared by issuing the following Unix commands:

```
sync
sync
echo 1 > /proc/sys/vm/drop_caches
```

Raw disk caches were cleared using the command:

```
dd of=/dev/null if=/dev/devicename count=1000 bs=1M
```

applied to each of the raw disk devices. Four of the queries – Q1 through Q4 – were basic spatial-temporal queries, requesting all the records that fell within a particular spatial-temporal "4D cube". These 4D cubes varied in both size and shape. Query Q5 requested all the records that fell inside a particular cube and that also satisfied conditions on the two other columns. The dimensions of the cubes for each of the queries are shown in the next table:

Query	**Time Bounds**	**Depth Bounds**	**Latitude Bounds**	**Longitude Bounds**
Q1	1,297,200 to 1,297,300 (small)	266.8 to 266.9 (small)	-35 to -25 (medium)	65 to 75 (medium)
Q2	1,297,200 to 1,297,300 (small)	266.8 to 266.9 (small)	-56.25 to -56.15 (small)	64.25 to 64.3 (small)
Q3	1,000,000 to 10,000,000(large)	100,000 to 400,000 (large)	-56.25 to -56.15 (small)	64.25 to 64.3 (small)
Q4	1,000,000 to 10,000,000 (large)	200 to 3,000 (medium)	-55 to -50 (medium)	65 to 70 (medium)
Q5	1,279,700 to 1,279,700 (small)	3,000 to 4,000,000 (large)	-35 to -15 (large)	70 to 80 (large)

Table 8: Query Cube Sizes and Shapes

For Q5, salinity was further restricted to values < -.00055 and temperature was further restricted to values > 27.5.

The number of rows returned by each of the queries, for each of the cases, is shown overleaf in Table 9. (Note the large and increasing numbers of rows returned in Q4.)

Case	Q1	Q2	Q3	Q4	Q5
55M_B_R	14,400	1	76	72,000	2
83M_B_R	14,400	1	114	108,000	2
110M_B_ R	14,400	1	152	144,000	2
138M_B_R	14,400	1	190	180,000	2
165M_B_R	14,400	1	228	216,000	2
193M_B_R	14,400	1	266	252,000	2
220M_B_R	14,400	1	304	288,000	2
440M_B_R	14,400	1	608	576,000	2

Table 9: Number of Rows Returned by Each Query

Native Informix Queries

For the native Informix cases the following SQL was run:

```
UNLOAD TO file SELECT
latitude,longitude,timeval,depth,temperature,salinity
    FROM  occam_conventional_hpl_ind_comp_${SIZE}
  WHERE latitude BETWEEN $MINLAT AND $MAXLAT AND
        longitude BETWEEN $MINLONG AND $MAXLONG AND
        timeval BETWEEN $MINDATE AND $MAXDATE AND
        depth BETWEEN $MINDEPTH AND $MAXDEPTH AND
        temperature > $MINTEMP AND salinity < $MAXSAL ;
```

No explicit query optimizer directives were provided, but the following SQL commands were issued in between loading the table and starting the queries:

```
UPDATE STATISTICS FOR TABLE occam_conventional_hpl_ind_comp_${SIZE};
UPDATE STATISTICS HIGH FOR TABLE
      occam_conventional_hpl_ind_comp_${SIZE}(timeval);
UPDATE STATISTICS HIGH FOR TABLE
      occam_conventional_hpl_ind_comp_${SIZE}(latitude);
UPDATE STATISTICS HIGH FOR TABLE
      occam_conventional_hpl_ind_comp_${SIZE}(longitude);
UPDATE STATISTICS HIGH FOR TABLE
      occam_conventional_hpl_ind_comp_${SIZE}(depth);
```

Executing these commands should allow the SQL query optimizer to select the best access path for executing the queries.

DBXten Queries

DBXten has two interfaces for querying data. The basic interface exposes the DSChip data type and the programmer has to convert DSChip's to conventional columns. The other uses Informix's Virtual Table Interface (VTI); it provides the illusion that a table of DSChip's contains conventional columns — the benefit being that it allows the `SELECT` statements in programs to remain virtually unchanged when DBXten is installed. Thus, VTI can enhance user/programmer productivity, while use of the basic interface usually leads to somewhat better performance in response times.

Basic Interface

The following command was used to perform the basic interface queries:

```
fetchData -table occam_chips_${SIZE}_0_tM -chip occamchip  \
      -boxcolumn 'occam_cube(occamchip)'  \
      -columns timeval,depth,latitude,longitude,temperature,salinity \
      -key "timeval,$MINDATE,$MAXDATE"  \
      -key "Latitude,$MINLAT,$MAXLAT"  \
      -key "Longitude,$MINLONG,$MAXLONG" \
      -key "Depth,$MINDEPTH,$MAXDEPTH" \
      -key "Temperature,$MINTEMP,100.0" \
      -key "Salinity,-1.0,$MAXSAL" >& file
```

Appendix C contains the `fetchData` program, written using the DBXten C API.

VTI Interface

The SQL below was used to create the virtual table:

```
        DROP TABLE occam_tempschemasource;
        CREATE TABLE occam_tempschemasource(schematext lvarchar);
        INSERT INTO occam_tempschemasource VALUES('(maxtuples
1000,timeval integer,depth double .01,latitude double .001,longitude
double .001,temperature double .01,salinity double .0000001)');

        DROP TABLE occam_chips_${size}_0_tM_vti;
        CREATE TABLE occam_chips_${size}_0_tM_vti(
           timeval integer,
           depth float,
           latitude float,
           longitude float,
           temperature float,
           salinity float)
        USING DSChipAccess(
             basetable='occam_chips_${size}_0_tm',
             dschipcolumn='occamchip',keylist=
      'occam_cube(occamchip):(timeval,latitude,longitude,depth)',
             schemasource='occam_tempschemasource',
             usescalarkeys='0');
```

The native Informix SQL shown earlier (modified to point to this virtual table) was then used to select data.

Query Time Comparison

The five tables on the next page contain the timing results for queries Q1 through Q5 in the various cases tested[5].

[5] The queries are defined in Table 8 on page 9.

Q1					
Case	Native	DBXten Basic	DBXten VTI	Basic Ratio	VTI Ratio
55M_B_R	00:05.5	00:01.2	00:02.2	0.225	0.408
83M_B_R	00:07.6	00:01.2	00:02.2	0.163	0.292
110M_B_ R	00:09.8	00:01.2	00:02.2	0.125	0.223
138M_B_R	00:12.0	00:01.2	00:02.2	0.099	0.181
165M_B_R	00:14.1	00:01.2	00:02.2	0.083	0.153
193M_B_R	00:16.4	00:01.2	00:02.2	0.072	0.133
220M_B_ R	00:18.4	00:01.3	00:02.2	0.072	0.117
440M_B_R	00:35.8	00:01.2	00:02.2	0.034	0.060

Q2					
Case	Native	DBXten Basic	DBXten VTI	Basic Ratio	VTI Ratio
55M_B_R	00:04.8	00:01.0	00:00.9	0.204	0.180
83M_B_R	00:07.0	00:01.0	00:00.9	0.145	0.127
110M_B_ R	00:09.2	00:01.0	00:00.9	0.114	0.098
138M_B_R	00:11.2	00:01.0	00:00.9	0.091	0.079
165M_B_R	00:13.3	00:01.0	00:00.9	0.073	0.067
193M_B_R	00:15.4	00:01.0	00:00.9	0.066	0.059
220M_B_ R	00:17.4	00:01.1	00:00.9	0.061	0.050
440M_B_R	00:33.9	00:01.0	00:00.9	0.031	0.027

Q3					
Case	Native	DBXten Basic	DBXten VTI	Basic Ratio	VTI Ratio
55M_B_R	00:04.9	00:01.9	00:01.8	0.387	0.367
83M_B_R	00:07.0	00:02.4	00:02.3	0.350	0.323
110M_B_ R	00:09.2	00:02.9	00:03.2	0.316	0.351
138M_B_R	00:11.2	00:03.4	00:03.1	0.301	0.280
165M_B_R	00:13.3	00:04.0	00:03.7	0.297	0.276
193M_B_R	00:15.5	00:04.3	00:04.1	0.278	0.265
220M_B_ R	00:18.1	00:04.8	00:04.5	0.266	0.250
440M_B_R	00:33.8	00:09.4	00:08.6	0.278	0.256

Q4					
Case	Native	DBXten Basic	DBXten VTI	Basic Ratio	VTI Ratio
55M_B_R	00:23.6	00:02.1	00:07.3	0.089	0.312
83M_B_R	00:34.5	00:02.9	00:10.6	0.083	0.309
110M_B_ R	00:45.3	00:03.5	00:14.0	0.076	0.310
138M_B_R	01:00.5	00:04.1	00:17.4	0.068	0.288
165M_B_R	01:19.3	00:04.9	00:20.9	0.061	0.263
193M_B_R	01:30.5	00:05.4	00:24.3	0.059	0.268
220M_B_ R	01:41.3	00:06.0	00:27.9	0.060	0.275
440M_B_R	03:11.6	00:10.7	00:54.8	0.056	0.286

Q5					
Case	Native	DBXten Basic	DBXten VTI	Basic Ratio	VTI Ratio
55M_B_R	01:11.4	00:03.4	00:18.6	0.047	0.261
83M_B_R	01:10.7	00:03.5	00:19.2	0.050	0.272
110M_B_ R	01:10.5	00:03.5	00:19.3	0.049	0.273
138M_B_R	01:10.3	00:03.5	00:19.7	0.050	0.280
165M_B_R	01:11.8	00:03.4	00:18.9	0.047	0.263
193M_B_R	01:12.2	00:03.3	00:19.8	0.046	0.274
220M_B_ R	01:11.5	00:03.3	00:19.7	0.047	0.276
440M_B_R	01:12.1	00:03.7	00:19.5	0.051	0.270

Table 10: Query Timings

Table 10 on the previous page shows that DBXten, with either the basic interface or VTI, performed queries much faster than when using just native Informix.

The timings in Table 10 are summarized in the table and figure below as cumulative query times for all five queries, when using each of the three options on each of the table sizes:

Million rows	Native	DBXten Basic	DBXten VTI	Ratio-Basic	Ratio-VTI
55	01:50.1	00:09.6	00:30.9	0.087	0.280
83	02:06.7	00:11.1	00:35.2	0.087	0.278
110	02:24.1	00:12.1	00:39.6	0.084	0.275
138	02:45.2	00:13.2	00:43.3	0.080	0.262
165	03:11.8	00:14.3	00:46.5	0.075	0.242
193	03:29.9	00:15.2	00:51.2	0.072	0.244
220	03:46.7	00:16.6	00:55.2	0.073	0.243
440	06:07.3	00:26.0	01:26.0	0.071	0.234

Table 11: Cumulative Query Times

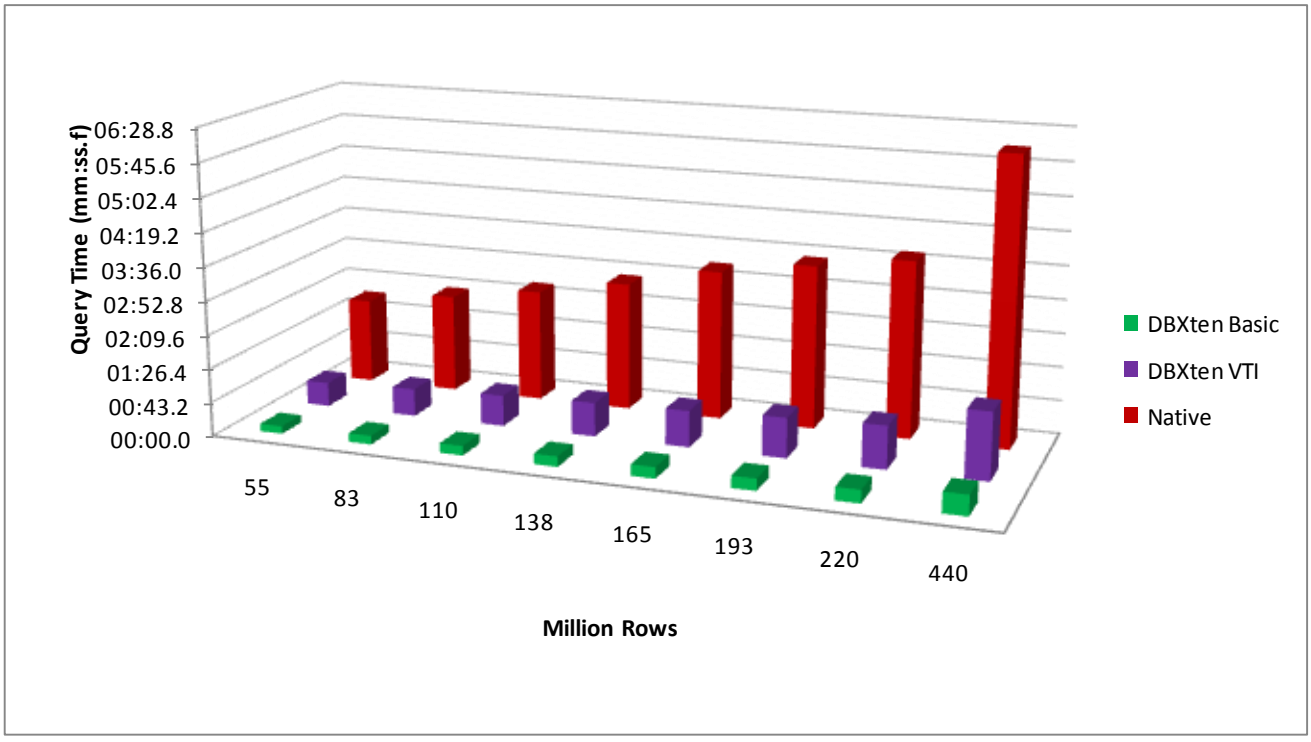

Figure 2: Cumulative Query Time as a Function of Table Size and Platform

The table and graphs above illustrate that the performance advantage of DBXten over native Informix improves as the data volumes increase.

Concurrent Query Testing

We conducted a final test to assess how well DBXten performs queries relative to native Informix under various load conditions. Each of queries Q1 through Q5 was run on the 220 million row table, and the numbers of concurrent users tested were N = 1, 2, 3, 5, 10, 15, 20, 25, and 30. The next table and graph present the results of this experiment:

N	Native-Avg	DBXten-Avg	Native-Max	DBXten-Max	Ratio-Avg
1	0:03:25.7	0:00:18.4	0:03:25.7	0:00:18.4	0.089
2	0:05:44.0	0:00:19.8	0:05:44.3	0:00:19.8	0.058
3	0:08:24.7	0:00:22.9	0:08:26.0	0:00:23.0	0.045
5	0:12:32.3	0:00:31.5	0:12:34.8	0:00:31.9	0.042
10	0:23:54.3	0:01:01.2	0:24:00.3	0:01:02.4	0.043
15	0:36:33.7	0:01:24.0	0:36:42.5	0:01:30.7	0.038
20	0:47:53.0	0:01:57.8	0:49:01.7	0:02:02.5	0.041
25	1:00:07.2	0:02:18.9	1:02:03.9	0:02:28.2	0.039
30	1:11:24.8	0:02:47.1	1:13:20.6	0:02:57.1	0.039

Table 12: Cumulative (Q1-Q5) Query Average and Maximum Times (h:mm:ss.f)

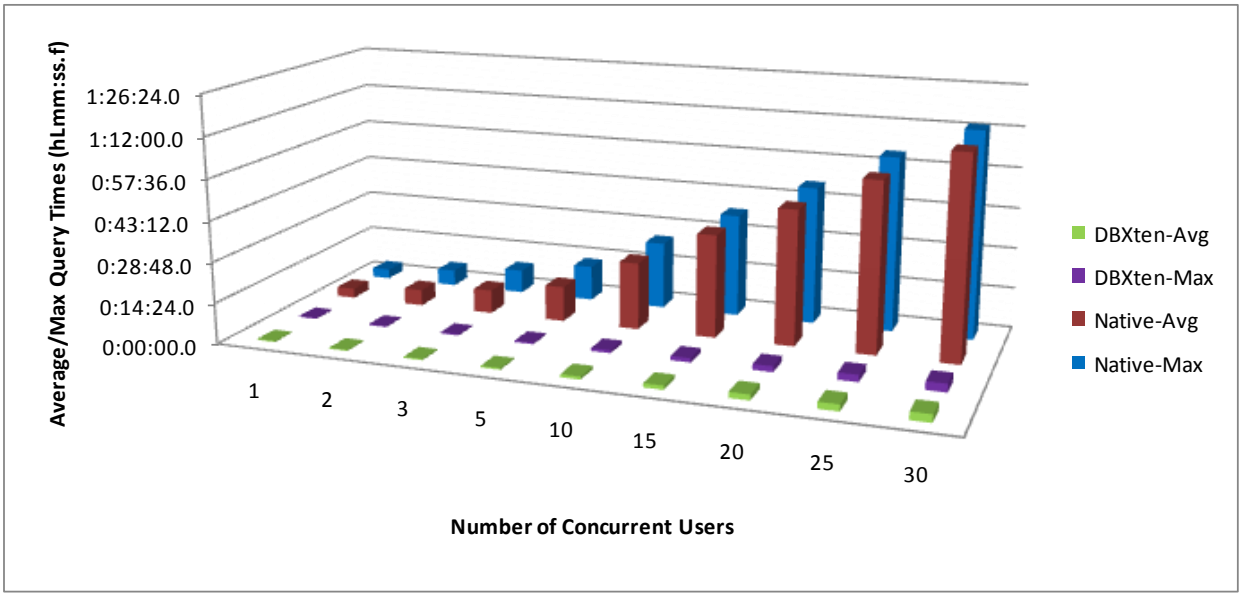

Figure 3: Cumulative (Q1-Q5) Query Average and Maximum Times (h:mm:ss.f) as a Function of Concurrency Level (N) and Platform.

These concurrency trials showed that Informix *with* DBXten provided answers to queries more than an order of magnitude faster than native Informix when multiple users were involved, and that this relative trend improved for DBXten as the number of concurrent users increased. Some additional remarks about these final query timings are probably in order. Note that for N = 1 in Table 12 the timings for native Informix and DBXten are slightly different than those shown in Table 11 corresponding to 220 million rows. This is

due to the way in which we performed the database restarts/cache clearing. In the non-concurrent tests we did this in between each query. This would not have been possible during concurrent testing, so instead the restarts/cache clearing was done just between each value of N. Also, different queries were performed by each of the N users[6] in order to minimize the effect of one user's query on another user's queries. However, the data read when answering a user's query was left in cache, thereby speeding up the same user's next query.

[6] The structure of the queries was the same but the specific bounds used were different, and, for any given user, the same bounds were used in the native Informix query as in the Informix with DBXten query.

Appendix A – onconfig File

```
#**************************************************************
# Licensed Material - Property Of IBM
#
# "Restricted Materials of IBM"
#
# IBM Informix Dynamic Server
# Copyright IBM Corporation 1996, 2008 All rights reserved.
#
# Title: onconfig.std
# Description: IBM Informix Dynamic Server Configuration Parameters
#
# Important: $INFORMIXDIR now resolves to the environment
# variable INFORMIXDIR. Replace the value of the INFORMIXDIR
# environment variable only if the path you want is not under
# $INFORMIXDIR.
#
# For additional information on the parameters:
# http://publib.boulder.ibm.com/infocenter/idshelp/v115/index.jsp
#**************************************************************

#**************************************************************
# Root Dbspace Configuration Parameters
#**************************************************************
# ROOTNAME     - The root dbspace name to contain reserved pages and
#                internal tracking tables.
# ROOTPATH     - The path for the device containing the root dbspace
# ROOTOFFSET   - The offset, in KB, of the root dbspace into the
#                device. The offset is required for some raw devices.
# ROOTSIZE     - The size of the root dbspace, in KB.  The value of
#                200000 allows for a default user space of about
#                100 MB and the default system space requirements.
# MIRROR       - Enable (1) or disable (0) mirroring
# MIRRORPATH   - The path for the device containing the mirrored
#                root dbspace
# MIRROROFFSET - The offset, in KB, into the mirrored device
#
# Warning: Always verify ROOTPATH before performing
#          disk initialization (oninit -i or -iy) to
#          avoid disk corruption of another instance
#**************************************************************

ROOTNAME rootdbs
ROOTOFFSET 0
MIRROR 0
MIRRORPATH /opt/IBM/informix/tmp/demo_on.root_mirror
MIRROROFFSET 0

#**************************************************************
# Physical Log Configuration Parameters
#**************************************************************
# PHYSFILE           - The size, in KB, of the physical log on disk.
#                      If RTO_SERVER_RESTART is enabled, the
#                      suggested formula for the size of PHSYFILE
#                      (up to about 1 GB) is:
```

```
#                           PHYSFILE = Size of BUFFERS * 1.1
# PLOG_OVERFLOW_PATH - The directory for extra physical log files
#                      if the physical log overflows during recovery
#                      or long transaction rollback
# PHYSBUFF           - The size of the physical log buffer, in KB
#*****************************************************************

PHYSFILE        300000
PLOG_OVERFLOW_PATH  /opt/IBM/informix/tmp
PHYSBUFF 128

#*****************************************************************
# Logical Log Configuration Parameters
#*****************************************************************
# LOGFILES     - The number of logical log files
# LOGSIZE      - The size of each logical log, in KB
# DYNAMIC_LOGS - The type of dynamic log allocation.
#                Acceptable values are:
#                2 Automatic. IDS adds a new logical log to the
#                  root dbspace when necessary.
#                1 Manual. IDS notifies the DBA to add new logical
#                  logs when necessary.
#                0 Disabled
# LOGBUFF      - The size of the logical log buffer, in KB
#*****************************************************************

LOGFILES        66
LOGSIZE 10000
DYNAMIC_LOGS 2
LOGBUFF 64

#*****************************************************************
# Long Transaction Configuration Parameters
#*****************************************************************
# If IDS cannot roll back a long transaction, the server hangs
# until more disk space is available.
#
# LTXHWM       - The percentage of the logical logs that can be
#                filled before a transaction is determined to be a
#                long transaction and is rolled back
# LTXEHWM      - The percentage of the logical logs that have been
#                filled before the server suspends all other
#                transactions so that the long transaction being
#                rolled back has exclusive use of the logs
#
# When dynamic logging is on, you can set higher values for
# LTXHWM and LTXEHWM because the server can add new logical logs
# during long transaction rollback. Set lower values to limit the
# number of new logical logs added.
#
# If dynamic logging is off, set LTXHWM and LTXEHWM to
# lower values, such as 50 and 60 or lower, to prevent long
# transaction rollback from hanging the server due to lack of
# logical log space.
#
# When using Enterprise Replication, set LTXEHWM to at least 30%
# higher than LTXHWM to minimize log overruns.
```

```
#########################################################################

LTXHWM 70
LTXEHWM 80

#########################################################################
# Server Message File Configuration Parameters
#########################################################################
# MSGPATH      - The path of the IDS message log file
# CONSOLE      - The path of the IDS console message file
#########################################################################

CONSOLE /work1/stat/bcslinuxtest_console.log

#########################################################################
# Tblspace Configuration Parameters
#########################################################################
# TBLTBLFIRST    - The first extent size, in KB, for the tblspace
#                  tblspace. Must be in multiples of the page size.
# TBLTBLNEXT     - The next extent size, in KB, for the tblspace
#                  tblspace. Must be in multiples of the page size.
# The default setting for both is 0, which allows IDS to manage
# extent sizes automatically.
#
# TBLSPACE_STATS - Enables (1) or disables (0) IDS to maintain
#                  tblspace statistics
#########################################################################

TBLTBLFIRST 0
TBLTBLNEXT 0
TBLSPACE_STATS 1

#########################################################################
# Temporary dbspace and sbspace Configuration Parameters
#########################################################################
# DBSPACETEMP  - The list of dbspaces used to store temporary
#                tables and other objects. Specify a colon
#                separated list of dbspaces that exist when the
#                server is started. If no dbspaces are specified,
#                or if all specified dbspaces  are not valid,
#                temporary files are created in the /tmp directory
#                instead.
# SBSPACETEMP  - The list of sbspaces used to store temporary
#                tables for smart large objects. If no sbspace
#                is specified, temporary files are created in
#                a standard sbspace.
#########################################################################

DBSPACETEMP temp1db
SBSPACETEMP temp1sb

#########################################################################
# Dbspace and sbspace Configuration Parameters
#########################################################################
# SBSPACENAME    - The default sbspace name where smart large objects
#                  are stored if no sbspace is specified during
#                  smart large object creation. Some DataBlade
```

```
#                   modules store smart large objects in this
#                   location.
# SYSSBSPACENAME - The default sbspace for system statistics
#                   collection. Otherwise, IDS stores statistics
#                   in the sysdistrib system catalog table.
# ONDBSPACEDOWN  - Specifies how IDS behaves when it encounters a
#                   dbspace that is offline. Acceptable values
#                   are:
#                   0 Continue
#                   1 Stop
#                   2 Wait for DBA action
#*****************************************************************

SBSPACENAME temp1sb
SYSSBSPACENAME temp1sbifmx
ONDBSPACEDOWN 2

#*****************************************************************
# System Configuration Parameters
#*****************************************************************
# SERVERNUM       - The unique ID for the IDS instance. Acceptable
#                   values are 0 through 255, inclusive.
# DBSERVERNAME    - The name of the default database server
# DBSERVERALIASES - The list of up to 32 alternative dbservernames,
#                   separated by commas
#*****************************************************************

#*****************************************************************
# Network Configuration Parameters
#*****************************************************************
# NETTYPE                     - The configuration of poll threads
#                               for a specific protocol. The
#                               format is:
#                               NETTYPE <protocol>,<# poll threads>
#                               ,<number of connections/thread>
#                               ,(NET|CPU)
#                               You can include multiple NETTYPE
#                               entries for multiple protocols.
# LISTEN_TIMEOUT              - The number of seconds that IDS
#                               waits for a connection
# MAX_INCOMPLETE_CONNECTIONS - The maximum number of incomplete
#                               connections before IDS logs a Denial
#                               of Service (DoS) error
# FASTPOLL                    - Enables (1) or disables (0) fast
#                               polling of your network, if your
#                               operating system supports it.
#*****************************************************************

NETTYPE ipcshm,1,50,CPU
LISTEN_TIMEOUT 60
MAX_INCOMPLETE_CONNECTIONS 1024
FASTPOLL 1

#*****************************************************************
# CPU-Related Configuration Parameters
#*****************************************************************
# MULTIPROCESSOR      - Specifies whether the computer has multiple
```

```
#                       CPUs. Acceptable values are: 0 (single
#                       processor), 1 (multiple processors or
#                       multi-core chips)
# VPCLASS cpu         - Configures the CPU VPs. The format is:
#                       VPCLASS cpu,num=<#>[,max=<#>][,aff=<#>]
#                       [,noage]
# VP_MEMORY_CACHE_KB  - Specifies the amount of private memory
#                       blocks of your CPU VP, in KB, that the
#                       database server can access.
#                       Acceptable values are:
#                       0 (disable)
#                       800 through 40% of the value of SHMTOTAL
# SINGLE_CPU_VP       - Optimizes performance if IDS runs with
#                       only one CPU VP. Acceptable values are:
#                       0 multiple CPU VPs
#                       Any nonzero value (optimize for one CPU VP)
##############################################################################

MULTIPROCESSOR 0
VP_MEMORY_CACHE_KB 0
SINGLE_CPU_VP 0

##############################################################################
# AIO and Cleaner-Related Configuration Parameters
##############################################################################
# VPCLASS aio  - Configures the AIO VPs. The format is:
#                VPCLASS aio,num=<#>[,max=<#>][,aff=<#>][,noage]
# CLEANERS     - The number of page cleaner threads
# AUTO_AIOVPS  - Enables (1) or disables (0) automatic management
#                of AIO VPs
# DIRECT_IO    - Enables (1) or disables (0) direct I/O for chunks
##############################################################################

#VPCLASS aio,num=1
CLEANERS 8
AUTO_AIOVPS 1
DIRECT_IO 0
#DIRECT_IO 1    -- seems to cause problem with compression???

##############################################################################
# Lock-Related Configuration Parameters
##############################################################################
# LOCKS              - The initial number of locks when IDS starts.
#                      Dynamic locking can add extra locks if needed.
# DEF_TABLE_LOCKMODE - The default table lock mode for new tables.
#                      Acceptable values are ROW and PAGE (default).
##############################################################################

LOCKS 20000
DEF_TABLE_LOCKMODE page

##############################################################################
# Shared Memory Configuration Parameters
##############################################################################
# RESIDENT         - Controls whether shared memory is resident.
#                    Acceptable values are:
#                    0 off (default)
```

```
#                     1 lock the resident segment only
#                     n lock the resident segment and the next n-1
#                        virtual segments, where n < 100
#                     -1 lock all resident and virtual segments
# SHMBASE           - The shared memory base address; do not change
# SHMVIRTSIZE       - The initial size, in KB, of the virtual
#                     segment of shared memory
# SHMADD            - The size, in KB, of additional virtual shared
#                     memory segments
# EXTSHMADD         - The size, in KB, of each extension shared
#                     memory segment
# SHMTOTAL          - The maximum amount of shared memory for IDS,
#                     in KB. A 0 indicates no specific limit.
# SHMVIRT_ALLOCSEG - Controls when IDS adds a memory segment and
#                     the alarm level if the memory segment cannot
#                     be added.
#                     For the first field, acceptable values are:
#                     - 0 Disabled
#                     - A decimal number indicating the percentage
#                       of memory used before a segment is added
#                     - The number of KB remaining when a segment
#                       is added
#                     For the second field, specify an alarm level
#                     from 1 (non-event) to 5 (fatal error).
# SHMNOACCESS       - A list of up to 10 memory address ranges
#                     that IDS cannot use to attach shared memory.
#                     Each address range is the start and end memory
#                     address in hex format, separated by a hyphen.
#                     Use a comma to separate each range in the list.
#########################################################################

RESIDENT 0
SHMBASE 0x44000000L
SHMVIRTSIZE 32656
SHMADD 8192
EXTSHMADD 8192
SHMTOTAL 0
SHMVIRT_ALLOCSEG 0,3
SHMNOACCESS

#########################################################################
# Checkpoint and System Block Configuration Parameters
#########################################################################
# CKPINTVL            - Specifies how often, in seconds, IDS checks
#                       if a checkpoint is needed. 0 indicates that
#                       IDS does not check for checkpoints. Ignored
#                       if RTO_SERVER_RESTART is set.
# AUTO_CKPTS          - Enables (1) or disables (0) monitoring of
#                       critical resource to trigger checkpoints
#                       more frequently if there is a chance that
#                       transaction blocking might occur.
# RTO_SERVER_RESTART - Specifies, in seconds, the Recovery Time
#                       Objective for IDS restart after a server
#                       failure. Acceptable values are 0 (off) and
#                       any number from 60-1800, inclusive.
# BLOCKTIMEOUT        - Specifies the amount of time, in seconds,
#                       for a system block.
```

```
#####################################################################

CKPTINTVL 300
AUTO_CKPTS 1
RTO_SERVER_RESTART 0
BLOCKTIMEOUT 3600

#####################################################################
# Transaction-Related Configuration Parameters
#####################################################################
# TXTIMEOUT        - The distributed transaction timeout, in seconds
# DEADLOCK_TIMEOUT - The maximum time, in seconds, to wait for a
#                    lock in a distributed transaction.
# HETERO_COMMIT    - Enables (1) or disables (0) heterogeneous
#                    commits for a distributed transaction
#                    involving an EGM gateway.
#####################################################################

TXTIMEOUT 300
DEADLOCK_TIMEOUT 60
HETERO_COMMIT 0

#####################################################################
# ontape Tape Device Configuration Parameters
#####################################################################
# TAPEDEV      - The tape device path for backups. To use standard
#                I/O instead of a device, set to stdio.
# TAPEBLK      - The tape block size, in KB, for backups
# TAPESIZE     - The maximum amount of data to put on one backup
#                tape. Acceptable values are 0 (unlimited) or any
#                positive integral multiple of TAPEBLK.
#####################################################################

TAPEBLK 32
TAPESIZE 0

#####################################################################
# ontape Logial Log Tape Device Configuration Parameters
#####################################################################
# LTAPEDEV     - The tape device path for logical logs
# LTAPEBLK     - The tape block size, in KB, for backing up logical
#                logs
# LTAPESIZE    - The maximum amount of data to put on one logical
#                log tape. Acceptable values are 0 (unlimited) or any
#                positive integral multiple of LTAPEBLK.
#####################################################################

LTAPEBLK 32
LTAPESIZE 0

#####################################################################
# Backup and Restore Configuration Parameters
#####################################################################
# BAR_ACT_LOG         - The ON-Bar activity log file location.
#                       Do not use the /tmp directory. Use a
#                       directory with restricted permissions.
# BAR_DEBUG_LOG       - The ON-Bar debug log file location.
```

```
#                       Do not use the /tmp directory. Use a
#                       directory with restricted permissions.
# BAR_DEBUG           - The debug level for ON-Bar. Acceptable
#                       values are 0 (off) through 9 (high).
# BAR_MAX_BACKUP      - The number of backup threads used in a
#                       backup. Acceptable values are 0 (unlimited)
#                       or any positive integer.
# BAR_RETRY           - Specifies the number of time to retry a
#                       backup or restore operation before reporting
#                       a failure
# BAR_NB_XPORT_COUNT  - Specifies the number of data buffers that
#                       each onbar_d process uses to communicate
#                       with the database server
# BAR_XFER_BUF_SIZE   - The size, in pages, of each data buffer.
#                       Acceptable values are 1 through 15 for
#                       4 KB pages and 1 through 31 for 2 KB pages.
# RESTARTABLE_RESTORE - Enables ON-Bar to continue a backup after a
#                       failure. Acceptable values are OFF or ON.
# BAR_PROGRESS_FREQ   - Specifies, in minutes, how often progress
#                       messages are placed in the ON-Bar activity
#                       log. Acceptable values are: 0 (record only
#                       completion messages) or 5 and above.
# BAR_BSALIB_PATH     - The shared library for ON-Bar and the
#                       storage manager. The default value is
#                       $INFORMIXDIR/lib/ibsad001 (with a
#                       platform-specific file extension).
# BACKUP_FILTER       - Specifies the pathname of a filter program
#                       to transform data during a backup, plus any
#                       program options
# RESTORE_FILTER      - Specifies the pathname of a filter program
#                       to transform data during a restore, plus any
#                       program options
# BAR_PERFORMANCE     - Specifies the type of performance statistics
#                       to report to the ON-Bar activity log for backup
#                       and restore operations.
#                       Acceptable values are:
#                        0 = Turn off performance monitoring (Default)
#                        1 = Display the time spent transferring data
#                            between the IDS instance and the storage
#                            manager
#                        2 = Display timestamps in microseconds
#                        3 = Display both timestamps and transfer
#                            statistics
##############################################################################

BAR_DEBUG 0
BAR_MAX_BACKUP 0
BAR_RETRY 1
BAR_NB_XPORT_COUNT 20
BAR_XFER_BUF_SIZE 31
RESTARTABLE_RESTORE ON
BAR_PROGRESS_FREQ 0
BAR_BSALIB_PATH
BACKUP_FILTER
RESTORE_FILTER
BAR_PERFORMANCE 0
```

```
#########################################################################
# Informix Storage Manager (ISM) Configuration Parameters
#########################################################################
# ISM_DATA_POOL - Specifies the name for the ISM data pool
# ISM_LOG_POOL  - Specifies the name for the ISM log pool
#########################################################################

ISM_DATA_POOL ISMData
ISM_LOG_POOL ISMLogs

#########################################################################
# Data Dictionary Cache Configuration Parameters
#########################################################################
# DD_HASHSIZE  - The number of data dictionary pools. Set to any
#                positive integer; a prime number is recommended.
# DD_HASHMAX   - The number of entries per pool.
#                Set to any positive integer.
#########################################################################

DD_HASHSIZE 31
DD_HASHMAX  10

#########################################################################
# Data Distribution Configuration Parameters
#########################################################################
# DS_HASHSIZE  - The number of data Ddstribution pools.
#                Set to any positive integer; a prime number is
#                recommended.
# DS_POOLSIZE  - The maximum number of entries in the data
#                distribution cache. Set to any positive integer.
#########################################################################

DS_HASHSIZE 31
DS_POOLSIZE 127

#########################################################################
# User Defined Routine (UDR) Cache Configuration Parameters
#########################################################################
# PC_HASHSIZE  - The number of UDR pools. Set to any
#                positive integer; a prime number is recommended.
# PC_POOLSIZE  - The maximum number of entries in the
#                UDR cache. Set to any positive integer.
#########################################################################

PC_HASHSIZE 31
PC_POOLSIZE 127

#########################################################################
# SQL Statement Cache Configuration Parameters
#########################################################################
# STMT_CACHE         - Controls SQL statement caching. Acceptable
#                      values are:
#                      0 Disabled
#                      1 Enabled at the session level
#                      2 All statements are cached
# STMT_CACHE_HITS    - The number of times an SQL statement must be
#                      executed before becoming fully cached.
```

```
#                       0 indicates that all statements are
#                       fully cached the first time.
# STMT_CACHE_SIZE    - The size, in KB, of the SQL statement cache
# STMT_CACHE_NOLIMIT - Controls additional memory consumption.
#                       Acceptable values are:
#                       0 Limit memory to STMT_CACHE_SIZE
#                       1 Obtain as much memory, temporarily, as needed
# STMT_CACHE_NUMPOOL - The number of pools for the SQL statement
#                       cache. Acceptable value is a positive
#                       integer between 1 and 256, inclusive.
#########################################################################

STMT_CACHE 0
STMT_CACHE_HITS 0
STMT_CACHE_SIZE 512
STMT_CACHE_NOLIMIT 0
STMT_CACHE_NUMPOOL 1

#########################################################################
# Operating System Session-Related Configuration Parameters
#########################################################################
# USEOSTIME        - The precision of SQL statement timing.
#                    Accepted values are 0 (precision to seconds)
#                    and 1 (precision to subseconds). Subsecond
#                    precision can degrade performance.
# STACKSIZE        - The size, in KB, for a session stack
# ALLOW_NEWLINE    - Controls whether embedded new line characters
#                    in string literals are allowed in SQL
#                    statements. Acceptable values are 1 (allowed)
#                    and any number other than 1 (not allowed).
# USELASTCOMMITTED - Controls the committed read isolation level.
#                    Acceptable values are:
#                    - NONE Waits on a lock
#                    - DIRTY READ Uses the last committed value in
#                      place of a dirty read
#                    - COMMITTED READ Uses the last committed value
#                      in place of a committed read
#                    - ALL Uses the last committed value in place
#                      of all isolation levels that support the last
#                      committed option
#########################################################################

USEOSTIME 0
STACKSIZE 32
ALLOW_NEWLINE 0
USELASTCOMMITTED NONE

#########################################################################
# Index Related Configuration Parameters
#########################################################################
# FILLFACTOR          - The percentage of index page fullness
# MAX_FILL_DATA_PAGES - Enables (1) or disables (0) filling data
#                       pages that have variable length rows as
#                       full as possible
# BTSCANNER           - Specifies the configuration settings for all
#                       btscanner threads. The format is:
#                       alice=(0-12)
```

```
# ONLIDX_MAXMEM       - The amount of memory, in KB, allocated for
#                       the pre-image pool and updator log pool for
#                       each partition.
#------------------------------------------------------------------------

FILLFACTOR 90
MAX_FILL_DATA_PAGES 0
ONLIDX_MAXMEM 5120

#------------------------------------------------------------------------
# Parallel Database Query (PDQ) Configuration Parameters
#------------------------------------------------------------------------
# MAX_PDQPRIORITY     - The maximum amount of resources, as a
#                       percentage, that PDQ can allocate to any
#                       one decision support query
# DS_MAX_QUERIES      - The maximum number of concurrent decision
#                       support queries
# DS_TOTAL_MEMORY     - The maximum amount, in KB, of decision
#                       support query memory
# DS_MAX_SCANS        - The maximum number of concurrent decision
#                       support scans
# DS_NONPDQ_QUERY_MEM - The amount of non-PDQ query memory, in KB.
#                       Acceptable values are 128 to 25% of
#                       DS_TOTAL_MEMORY.
# DATASKIP            - Specifies whether to skip dbspaces when
#                       processing a query. Acceptable values are:
#                       - ALL Skip all unavailable fragments
#                       - ON <dbspace1> <dbspace2>... Skip listed
#                         dbspaces
#                       - OFF Do not skip dbspaces (default)
#------------------------------------------------------------------------

MAX_PDQPRIORITY 100
DS_MAX_QUERIES
DS_TOTAL_MEMORY
DS_MAX_SCANS 1048576
DS_NONPDQ_QUERY_MEM 128
DATASKIP

#------------------------------------------------------------------------
# Optimizer Configuration Parameters
#------------------------------------------------------------------------
# OPTCOMPIND     - Controls how the optimizer determines the best
#                  query path. Acceptable values are:
#                  0 Nested loop joins are preferred
#                  1 If isolation level is repeatable read,
#                    works the same as 0, otherwise works same as 2
#                  2 Optimizer decisions are based on cost only
# DIRECTIVES     - Specifies whether optimizer directives are
#                  enabled (1) or disabled (0). Default is 1.
# EXT_DIRECTIVES - Controls the use of external SQL directives.
#                  Acceptable values are:
#                  0 Disabled
#                  1 Enabled if the IFX_EXT_DIRECTIVES environment
#                     variable is enabled
#                  2 Enabled even if the IFX_EXT_DIRECTIVES
#                     environment is not set
```

```
# OPT_GOAL       - Controls how the optimizer should optimize for
#                  fastest retrieval. Acceptable values are:
#                  -1 All rows in a query
#                  0 The first rows in a query
# IFX_FOLDVIEW   - Enables (1) or disables (0) folding views that
#                  have multiple tables or a UNION ALL clause.
#                  Disabled by default.
# AUTO_REPREPARE - Enables (1) or disables (0) automatically
#                  re-optimizing stored procedures and re-preparing
#                  prepared statements when tables that are referenced
#                  by them change. Minimizes the occurrence of the
#                  -710 error.
#########################################################################

OPTCOMPIND 2
DIRECTIVES 1
EXT_DIRECTIVES 0
OPT_GOAL -1
IFX_FOLDVIEW 0
AUTO_REPREPARE 1

#########################################################################
# Read-ahead Configuration Parameters
#########################################################################
#RA_PAGES      - The number of pages, as a positive integer, to
#                attempt to read ahead
#RA_THRESHOLD  - The number of pages, as a postive integer, left
#                before the next read-ahead group
#########################################################################

RA_PAGES     64
RA_THRESHOLD 16

#########################################################################
# SQL Tracing and EXPLAIN Plan Configuration Parameters
#########################################################################
# EXPLAIN_STAT - Enables (1) or disables (0) including the Query
#                Statistics section in the EXPLAIN output file
# SQLTRACE     - Configures SQL tracing. The format is:
#                mode=(global|user)
#########################################################################

EXPLAIN_STAT 0

#########################################################################
# Security Configuration Parameters
#########################################################################
# DBCREATE_PERMISSION        - Specifies the users who can create
#                              databases (by default, any user can).
#                              Add a DBCREATE_PERMISSION entry
#                              for each user who needs database
#                              creation privileges. Ensure user
#                              informix is authorized when you
#                              first initialize IDS.
# DB_LIBRARY_PATH            - Specifies the locations, separated
#                              by commas, from which IDS can use
#                              UDR or UDT shared libraries. If set,
```

```
#                               make sure that all directories
containing
#                               the blade modules are listed, to
#                               ensure all DataBlade modules will
#                               work.
# IFX_EXTEND_ROLE             - Controls whether administrators
#                               can use the EXTEND role to specify
#                               which users can register external
#                               routines. Acceptable values are:
#                               0 Any user can register external
#                                  routines
#                               1 Only users granted the ability
#                                 to register external routines
#                                 can do so (Default)
# SECURITY_LOCALCONNECTION    - Specifies whether IDS performs
#                               security checking for local
#                               connections. Acceptable values are:
#                               0 Off
#                               1 Validate ID
#                               2 Validate ID and port
# UNSECURE_ONSTAT             - Controls whether non-DBSA users are
#                               allowed to run all onstat commands.
#                               Acceptable values are:
#                               1 Enabled
#                               0 Disabled (Default)
# ADMIN_USER_MODE_WITH_DBSA   - Controls who can connect to IDS
#                               in administration mode. Acceptable
#                               values are:
#                               1 DBSAs, users specified by
#                                  ADMIN_MODE_USERS, and the user
#                                  informix
#                               0 Only the user informix (Default)
# ADMIN_MODE_USERS            - Specifies the user names, separated by
#                               commas, who can connect to IDS in
#                               administration mode, in addition to
#                               the user informix
# SSL_KEYSTORE_LABEL          - The label, up to 512 characters, of
#                               the IDS certificate used in Secure
#                               Sockets Layer (SSL) protocol
#                               communications.
##############################################################################

#DBCREATE_PERMISSION informix
#DB_LIBRARY_PATH
IFX_EXTEND_ROLE 0
SECURITY_LOCALCONNECTION
UNSECURE_ONSTAT
ADMIN_USER_MODE_WITH_DBSA
ADMIN_MODE_USERS
SSL_KEYSTORE_LABEL
##############################################################################
# LBAC Configuration Parameters
##############################################################################
# PLCY_POOLSIZE - The maximum number of entries in each hash
#                 bucket of the LBAC security information cache
# PLCY_HASHSIZE - The number of hash buckets in the LBAC security
#                 information cache
```

```
# USRC_POOLSIZE - The maximum number of entries in each hash
#                 bucket of the LBAC credential memory cache
# USRC_HASHSIZE - The number of hash buckets in the LBAC credential
#                 memory cache
#####################################################################

PLCY_POOLSIZE 127
PLCY_HASHSIZE 31
USRC_POOLSIZE 127
USRC_HASHSIZE 31

#####################################################################
# Optical Configuration Parameters
#####################################################################
# STAGEBLOB    - The name of the optical blobspace. Must be set to
#                use the optical-storage subsystem.
# OPCACHEMAX   - Maximum optical cache size, in KB
#####################################################################

STAGEBLOB
OPCACHEMAX 0

#####################################################################
# High Availability and Enterprise Replication Security
# Configuration Parameters
#####################################################################
# ENCRYPT_HDR     - Enables (1) or disables (0) encryption for HDR.
# ENCRYPT_SMX     - Controls the level of encryption for RSS and
#                   SDS servers. Acceptable values are:
#                   0 Do not encrypt (Default)
#                   1 Encrypt if possible
#                   2 Always encrypt
# ENCRYPT_CDR     - Controls the level of encryption for ER.
#                   Acceptable values are:
#                   0 Do not encrypt (Default)
#                   1 Encrypt if possible
#                   2 Always encrypt
# ENCRYPT_CIPHERS - A list of encryption ciphers and modes,
#                   separated by commas. Default is all.
# ENCRYPT_MAC     - Controls the level of message authentication
#                   code (MAC). Acceptable values are off, high,
#                   medium, and low. List multiple values separated
#                   by commas; the highest common level between
#                   servers is used.
# ENCRYPT_MACFILE - The paths of the MAC key files, separated
#                   by commas. Use the builtin keyword to specify
#                   the built-in key. Default is builtin.
# ENCRYPT_SWITCH  - Defines the frequencies, in minutes, at which
#                   ciphers and keys are renegotiated. Format is:
#                   <cipher_switch_time>,<key_switch_time>
#                   Default is 60,60.
#####################################################################

ENCRYPT_HDR
ENCRYPT_SMX
ENCRYPT_CDR 0
ENCRYPT_CIPHERS
```

```
ENCRYPT_MAC
ENCRYPT_MACFILE
ENCRYPT_SWITCH

#*********************************************************************
# Enterprise Replication (ER) Configuration Parameters
#*********************************************************************
# CDR_EVALTHREADS         - The number of evaluator threads per
#                           CPU VP and the number of additional
#                           threads, separated by a comma.
#                           Acceptable values are: a non-zero value
#                           followed by a non-negative value
# CDR_DSLOCKWAIT          - The number of seconds the Datasync
#                           waits for database locks.
# CDR_QUEUEMEM            - The maximum amount of memory, in KB,
#                           for the send and receive queues.
# CDR_NIFCOMPRESS         - Controls the network interface
#                           compression level.
#                           Acceptable values are:
#                           -1 Never
#                           0 None
#                           1-9 Compression level
# CDR_SERIAL              - Specifies the incremental size and
#                           the starting value of replicated
#                           serial columns. The format is:
#                           <delta>,<offset>
# CDR_DBSPACE             - The dbspace name for the syscdr
#                           database.
# CDR_QHDR_DBSPACE        - The name of the transaction record
#                           dbspace. Default is the root dbspace.
# CDR_QDATA_SBSPACE       - The names of sbspaces for spooled
#                           transaction data, separated by commas.
# CDR_MAX_DYNAMIC_LOGS    - The maximum number of dynamic log
#                           requests that ER can make within one
#                           server session. Acceptable values are:
#                           -1 (unlimited), 0 (disabled),
#                           1 through n (limit to n requests)
# CDR_SUPPRESS_ATSRISWARN - The Datasync error and warning code
#                           numbers to be suppressed in ATS and RIS
#                           files. Acceptable values are: numbers
#                           or ranges of numbers separated by commas.
#                           Separate numbers in a range by a hyphen.
#*********************************************************************

CDR_EVALTHREADS 1,2
CDR_DSLOCKWAIT 5
CDR_QUEUEMEM 4096
CDR_NIFCOMPRESS 0
CDR_SERIAL 0
CDR_DBSPACE
CDR_QHDR_DBSPACE
CDR_QDATA_SBSPACE
CDR_MAX_DYNAMIC_LOGS 0
CDR_SUPPRESS_ATSRISWARN

#*********************************************************************
# High Availability Cluster (HDR, SDS, and RSS)
```

```
# Configuration Parameters
##############################################################################
# DRAUTO             - Controls automatic failover of primary
#                      servers. Valid for HDR, SDS, and RSS.
#                      Acceptable values are:
#                      0 Manual
#                      1 Retain server type
#                      2 Reverse server type
#                      3 Connection Manager Arbitrator controls
#                        server type
# DRINTERVAL         - The maximum interval, in seconds, between HDR
#                      buffer flushes. Valid for HDR only.
# DRTIMEOUT          - The time, in seconds, before a network
#                      timeout occurs. Valid for HDR only.
# DRLOSTFOUND        - The path of the HDR lost-and-found file.
#                      Valid of HDR only.
# DRIDXAUTO          - Enables (1) or disables (0) automatic index
#                      repair for an HDR pair. Default is 0.
# HA_ALIAS           - The server alias for a high-availability
#                      cluster. Must be the same as a value of
#                      DBSERVERNAME or DBSERVERALIASES that uses a
#                      network-based connection type. Valid for HDR,
#                      SDS, and RSS.
# LOG_INDEX_BUILDS   - Enable (1) or disable (0) index page logging.
#                      Required for RSS. Optional for HDR and SDS.
# SDS_ENABLE         - Enables (1) or disables (0) an SDS server.
#                      Set this value on an SDS server after setting
#                      up the primary. Valid for SDS only.
# SDS_TIMEOUT        - The time, in seconds, that the primary waits
#                      for an acknowledgement from an SDS server
#                      while performing page flushing before marking
#                      the SDS server as down. Valid for SDS only.
# SDS_TEMPDBS        - List of up to 16 temporary dbspaces used by an
#                      SDS server. The format for each dbspace entry is:
#                      <dbspace_name>,<path>,<pagesize>,<offset>,
#                      <offset_size>
#                      Separate entries with a space. Valid for SDS.
# SDS_PAGING         - The paths of two buffer paging files,
#                      Separated by a comma. Valid for SDS only.
# REDIRECTED_WRITES  - The number of SMX pipes for redirected writes
#                      between a primary and a secondary server. Valid
#                      for HDR, SDS, and RSS. Acceptable values are:
#                      0 Disable redirected writes
#                      1 through twice the number of CPU VPs
# FAILOVER_CALLBACK  - Specifies the path and program name called when a
#                      secondary server transitions to a standard or
#                      primary server. Valid for HDR, SDS, and RSS.
# TEMPTAB_NOLOG      - Controls the default logging mode for temporary
#                      tables that are explicitly created with the
#                      CREATE TEMP TABLE or SELECT INTO TEMP statements.
#                      Secondary servers must not have logged temporary
#                      tables. Acceptable values are:
#                      0 Create temporary tables with logging enabled by
#                        default.
#                      1 Create temporary tables without logging.
#                      Required to be set to 1 on HDR, RSS, and SDS
#                      secondary servers.
```

```
#
# Warning: Always verify ROOTPATH before performing
#          disk initialization (oninit -i or -iy) to
#          avoid disk corruption of another instance
#########################################################################

DRAUTO 0
DRINTERVAL 30
DRTIMEOUT 30
HA_ALIAS
DRIDXAUTO 0
LOG_INDEX_BUILDS
SDS_ENABLE
SDS_TIMEOUT 20
SDS_TEMPDBS
SDS_PAGING
REDIRECTED_WRITES 0
FAILOVER_CALLBACK
TEMPTAB_NOLOG 0

#########################################################################
# Logical Recovery Parameters
#########################################################################
# ON_RECVRY_THREADS  - The number of logical recovery threads that
#                      run in parallel during a warm restore.
# OFF_RECVRY_THREADS - The number of logical recovery threads used
#                      in a cold restore. Also, the number of
#                      threads used during fast recovery.
#########################################################################

ON_RECVRY_THREADS  1
OFF_RECVRY_THREADS 10

#########################################################################
# Diagnostic Dump Configuration Parameters
#########################################################################
# DUMPDIR      - The location Assertion Failure (AF) diagnostic
#                files
# DUMPSHMEM    - Controls shared memory dumps. Acceptable values
#                are:
#                0 Disabled
#                1 Dump all shared memory
#                2 Exclude the buffer pool from the dump
# DUMPGCORE    - Enables (1) or disables (0) whether IDS dumps a
#                core using gcore
# DUMPCORE     - Enables (1) or disables (0) whether IDS dumps a
#                core after an AF
# DUMPCNT      - The maximum number of shared memory dumps or
#                core files for a single session
#########################################################################

DUMPSHMEM 1
DUMPGCORE 0
DUMPCORE 0
DUMPCNT 1

#########################################################################
```

```
# Alarm Program Configuration Parameters
#########################################################################
# ALARMPROGRAM       - Specifies the alarm program to display event
#                      alarms. To enable automatic logical log backup,
#                      edit alarmprogram.sh and set BACKUPLOGS=Y.
# ALRM_ALL_EVENTS    - Controls whether the alarm program runs for
#                      every event. Acceptable values are:
#                      1 Logs only noteworthy events
#                      2 Logs all events
# STORAGE_FULL_ALARM - <time interval in seconds>,<alarm severity>
#                      specifies in what interval:
#                      - a message will be printed to the online.log
file
#                      - an alarm will be raised
#                      when
#                      - a dbspace becomes full
#                        (ISAM error -131)
#                      - a partition runs out of pages or extents
#                        (ISAM error -136)
#                      time interval = 0 : OFF
#                      severity = 0 : no alarm, only message
# SYSALARMPROGRAM    - Specifies the system alarm program triggered
#                      when an AF occurs
#########################################################################

ALRM_ALL_EVENTS 0
STORAGE_FULL_ALARM 600,3

#########################################################################
# RAS Configuration Parameters
#########################################################################
# RAS_PLOG_SPEED - Technical Support diagnostic parameter.
#                  Do not change; automatically updated.
# RAS_LLOG_SPEED - Technical Support diagnostic parameter.
#                  Do not change; automatically updated.
#########################################################################

RAS_PLOG_SPEED  13422
RAS_LLOG_SPEED  742

#########################################################################
# Character Processing Configuration Parameter
#########################################################################
# EILSEQ_COMPAT_MODE - Controls whether when processing characters,
#                      IDS checks if the characters are valid for
#                      the locale and returns error -202 if they are
#                      not. Acceptable values are:
#                      0 Return an error for characters that are not
#                        valid (Default)
#                      1 Allow characters that are not valid
#########################################################################

EILSEQ_COMPAT_MODE  0

#########################################################################
# Statistic Configuration Parameters
#########################################################################
```

```
# QSTATS  - Enables (1) or disables (0) the collection of queue
#           statistics that can be viewed with onstat -g qst
# WSTATS  - Enables (1) or disables (0) the collection of wait
#           statistics that can be viewed with onstat -g wst
#*****************************************************************

QSTATS 0
WSTATS 0
#*****************************************************************
# Java Configuration Parameters
#*****************************************************************
# VPCLASS jvp  - Configures the Java VP. The format is:
#                VPCLASS jvp,num=<#>[,max=<#>][,aff=<#>][,noage]
# JVPJAVAHOME  - The JRE root directory
# JVPHOME      - The Krakatoa installation directory
# JVPPROPFILE  - The Java VP property file
# JVPLOGFILE   - The Java VP log file
# JDKVERSION   - The version of JDK supported by this server
#                This parameter is deprecated and is no longer required
# JVPJAVALIB   - The location of the JRE libraries, relative
#                to JVPJAVAHOME
# JVPJAVAVM    - The JRE libraries to use for the Java VM
# JVPARGS      - Configures the Java VM. To display JNI calls,
#                use JVPARGS -verbose:jni. Separate options with
#                semicolons.
# JVPCLASSPATH - The Java classpath to use. Use krakatoa_g.jar
#                for debugging. Comment out the JVPCLASSPATH
#                entry you do not want to use.
#*****************************************************************

#VPCLASS        jvp,num=1
#JDKVERSION      1.5
JVPJAVALIB      /bin
JVPJAVAVM       jvm
#JVPARGS        -verbose:jni
#JVPCLASSPATH
/usr/informix/extend/krakatoa/krakatoa_g.jar:/usr/informix/extend/kraka
toa/jdbc_g.jar

#*****************************************************************
# Buffer pool and LRU Configuration Parameters
#*****************************************************************
# BUFFERPOOL      - Specifies the default values for buffers and LRU
#                   queues in each buffer pool. Each page size used
#                   by a dbspace has a buffer pool and needs a
#                   BUFFERPOOL entry. The onconfig.std file contains
#                   two initial entries: a default entry from which
#                   to base new page size entries on, and an entry
#                   for the operating system default page
size.
#                   When you add a dbspace with a different page size,
#                   IDS adds a BUFFERPOOL entry to the onconfig file
#                   with values that are the same as the default
#                   BUFFERPOOL entry, except that the default
#                   new page size. With interval checkpoints, these
#                   values can now be set higher than in previous
#                   versions of IDS in an OLTP environment.
```

```
# AUTO_LRU_TUNING - Enables (1) or disables (0) automatic tuning of
#                   LRU queues. When this parameter is enabled, IDS
#                   increases the LRU flushing if it cannot find low
#                   priority buffers for number page faults.
##############################################################################

AUTO_LRU_TUNING 1

ROOTPATH /usr/INFORMIXDATA/bcslinuxtest/rootdb
MSGPATH /work1/stat/bcslinuxtest_online.log
TAPEDEV /dev/null
LTAPEDEV /dev/null
DBSERVERNAME bcslinuxtest
DBSERVERALIASES
SERVERNUM 0
ALARMPROGRAM /opt/IBM/informix/etc/alarmprogram.sh
DRLOSTFOUND /opt/IBM/informix/etc/dr.lostfound
BAR_ACT_LOG /work1/stat/bar_act.log
BAR_DEBUG_LOG /work1/stat/bar_debug.log
SYSALARMPROGRAM /opt/IBM/informix/etc/evidence.sh
DUMPDIR /work1/stat
JVPJAVAHOME /opt/IBM/informix/extend/krakatoa/jre
JVPHOME /opt/IBM/informix/extend/krakatoa/
JVPPROPFILE /opt/IBM/informix/extend/krakatoa/.jvpprops
JVPLOGFILE /opt/IBM/informix/demo/server/jvp.log
JVPCLASSPATH
/opt/IBM/informix/extend/krakatoa/krakatoa.jar:/opt/IBM/informix/extend
/krakatoa/jdbc.jar
ROOTSIZE 200000
#BUFFERPOOL     size=2K,buffers=1000,lrus=8,lru_min_dirty=50.000000,lru
_max_dirty=60.000000
BUFFERPOOL      size=2K,buffers=100000,lrus=8,lru_min_dirty=50.000000,l
ru_max_dirty=60.000000
VPCLASS cpu,num=1,noage
VPCLASS dbxten,num=1,noyield
VPCLASS ufi,num=1,noyield
BTSCANNER       num=1,threshold=5000,rangesize=-1,alice=6
UPDATABLE_SECONDARY 0
BATCHEDREAD_TABLE 1
CONVERSION_GUARD 1
RESTORE_POINT_DIR /opt/IBM/informix.11.50_UC6/tmp
DELAY_APPLY     0
STOP_APPLY      0
LOG_STAGING_DIR
JVPLOGFILE      /opt/IBM/informix/extend/krakatoa//jvp.log
```

Appendix B – Abridged NetCDF File Header (ncdump)

```
dimensions:
        TIMESTEP = 1 ;
        LONGITUDE_T = 447 ;
        LATITUDE_T = 467 ;
        LONGITUDE_U = 447 ;
        LATITUDE_U = 467 ;
        DEPTH = 66 ;
        DEPTH_EDGES = 67 ;
variables:
        int TIMESTEP(TIMESTEP) ;
                TIMESTEP:long_name = "Timestep" ;
        float LONGITUDE_T(LONGITUDE_T) ;
                LONGITUDE_T:long_name = "Longitude" ;
                LONGITUDE_T:units = "degrees" ;
                LONGITUDE_T:Format = "F10.4" ;
        float LATITUDE_T(LATITUDE_T) ;
                LATITUDE_T:long_name = "Latitude" ;
                LATITUDE_T:units = "degrees" ;
                LATITUDE_T:Format = "F10.4" ;
        float DEPTH(DEPTH) ;
                DEPTH:long_name = "Depth" ;
                DEPTH:units = "cm" ;
                DEPTH:Format = "F5.2" ;
                DEPTH:positive = "down" ;
                DEPTH:edges = "DEPTH_EDGES" ;
        float POTENTIAL_TEMPERATURE__MEAN_(DEPTH, LATITUDE_T,
LONGITUDE_T) ;
                POTENTIAL_TEMPERATURE__MEAN_:long_name = "potential
temperature (mean)" ;
                POTENTIAL_TEMPERATURE__MEAN_:units = "C" ;
                POTENTIAL_TEMPERATURE__MEAN_:_FillValue = 0.f ;
                POTENTIAL_TEMPERATURE__MEAN_:LEVEL = 0 ;
                POTENTIAL_TEMPERATURE__MEAN_:T_GRID = -1 ;
        float SALINITY__MEAN_(DEPTH, LATITUDE_T, LONGITUDE_T) ;
                SALINITY__MEAN_:long_name = "salinity (mean)" ;
                SALINITY__MEAN_:units = "(PSU-35)/1000" ;
                SALINITY__MEAN_:_FillValue = 0.f ;
                SALINITY__MEAN_:LEVEL = 0 ;
                SALINITY__MEAN_:T_GRID = -1 ;
...

// global attributes:
                :parentfile =
"/scratch/arther3/occamp083/run401/monthly/nov2003.h5m1" ;
                :creation_command =
"/noc/users/acc/SUBVOL_UTILS/h52ncsubvol.novel -f
/scratch/arther3/occamp083/run401/monthly/nov2003.h5m1 -o /noc/om
f/scratch/lsm/javad/WORKING/4739/nov2003.h5m1subvol -include
\"POTENTIAL TEMPERATURE, SALINITY, U VELOCITY, V VELOCITY,\" -sw 726
264 -ne 1172 730 -k 1
 66 -stride 1 1 1 -domain 66 4320 1735" ;
                :FMODE = 2 ;
                :FTYPE = 2 ;
                :ROTATION = 1 ;
```

```
data:

 TIMESTEP = 2077464 ;

...
```

Appendix C – Source of fetchData Program

```
#include <math.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <DSError.h>
#include <genparseopt.h>
#include <dschip_const.h>
#include <dschip_exports.h>
#include <dschipInformixClient.h>

EXEC SQL include sqltypes;
EXEC SQL include exp_chk.ec;

static FILE *outfile;
static int verbose = 0;

static char *outFileName = NULL;
static char *tableName = NULL;
static char *chipColumnName = NULL;
static char *boxColumn = NULL;

#define maxCOLUMNS (40)
static int numColumns = 0;
static char *columns[maxCOLUMNS];

#define maxRANGES (40)
int numRanges = 0;
char *rangeColumn[maxRANGES];
char *rangeLow[maxRANGES];
char *rangeHigh[maxRANGES];
double rangeHighVal[maxRANGES];
double rangeLowVal[maxRANGES];
int rangeIsKey[maxRANGES];
int numKeys = 0;



static void SetBoxColumn(char *i)
{
   boxColumn = i;
}

static void SetChipName(char *i)
{
   chipColumnName = i;
}

static void SetFileName(char *i)
{
   outFileName = i;
}
```

```
static void SetTableName(char *i)
{
   tableName = i;
}

static void SetColumn(char *i)
{
   char *item;
   for(;;) {
      item = strtok(i, ",");
      if( !item ) break;

      columns[numColumns++]  = item;
      i = NULL;
   }
}

static double ParseScalar(char *strValue)
{
   if( strchr(strValue, '-') > strValue ) {
       return DSGMTStringToDouble(strValue);
   }
   else {
       double val;
       if( sscanf(strValue, "%lf", &val) != 1 ) {
          DSUserError("Bad numeric value '%s'", strValue);
       }
       return val;
   }
}

static void SetRangeCommon(char *i) {
    char buffer[255];
    char *name;
    char *lowStr;
    char *highStr;
    double lowVal, highVal;

    strncpy(buffer, i, sizeof(buffer));
    rangeColumn[numRanges] = strtok(i, ",");
    rangeLow[numRanges] = strtok(NULL, ",");
    rangeHigh[numRanges]= strtok(NULL, ",");
    if( !rangeHigh[numRanges]) {
       DSUserError("bad range '%s'", buffer);
    }
    rangeLowVal[numRanges] = ParseScalar(rangeLow[numRanges]);
    rangeHighVal[numRanges] = ParseScalar(rangeHigh[numRanges]);

    numRanges++;
}

static void SetRange(char *i){
    rangeIsKey[numRanges] = 0;
    SetRangeCommon(i);
}

static void SetKey(char *i) {
```

```
    rangeIsKey[numRanges] = 1;
    numKeys++;
    SetRangeCommon(i);
}

static void ProcessSingleChip(DSChip *chip)
{
   int chipRows, numVars;
   int i, j;
   DSChipVar *vars[maxCOLUMNS];
   DSChipVar *rangeVars[maxCOLUMNS];
   char format[maxCOLUMNS][10];
   int types[maxCOLUMNS];

   chipRows = DSChipGetNumRows(chip);
   for(i = 0; i < numColumns; i++ ) {
      vars[i] = DSChipGetVarByName(chip, columns[i]);
      if( vars[i] == NULL ) {
         if( verbose ) {
            fprintf(stderr, "skipping chip without column %s",
columns[i]);
         }
         return;
      }
      types[i] = DSChipVarGetType(vars[i]);
      if( types[i] == DSVarTypeDOUBLE ) {
         int fractionLength;
         double tolerance;
         tolerance = DSChipVarGetTolerance(vars[i]);
         if( tolerance == 0 ) {
             fractionLength = 14;
         }
         else if( tolerance >= 1 ) {
            fractionLength = 0;
         }
         else {
            fractionLength = (int)ceil(-log10(tolerance));
         }
         sprintf(format[i], "%%.%df", fractionLength);
      }
   }

   for(i = 0; i < numRanges; i++ ) {
      rangeVars[i] = DSChipGetVarByName(chip, rangeColumn[i]);
   }

   for( j = 0; j < chipRows; j++ ) {
       int isInside = 1;

       for( i = 0; i < numRanges; i++ ) {
           double val = DSChipVarGetDouble(rangeVars[i], j);
           if( val < rangeLowVal[i] || val > rangeHighVal[i] ) {
              isInside = 0;
              break;
           }
       }
       if( !isInside ) continue;
```

```
        for( i = 0; i < numColumns; i++ ) {
          if( i > 0 ) {
             fprintf(outfile, ",");
          }
          switch( types[i]) {
            case DSVarTypeDATE:
               {
                  char timeBuffer[80];
                  DSDoubleToGMTString(timeBuffer,
                         DSChipVarGetDouble(vars[i], j), 4);
                  fprintf(outfile, "%s", timeBuffer);
               }
               break;
            case DSVarTypeDOUBLE:
               fprintf(outfile, format[i], DSChipVarGetDouble(vars[i],
j));
               break;
            case DSVarTypeSTRING:
               fprintf(outfile, "%s",
                  DSChipVarGetString(vars[i], j));
               break;
            case DSVarTypeINT:
               fprintf(outfile, "%d",
                  DSChipVarGetInt(vars[i], j));
               break;
          }
        }
        fprintf(outfile, "\n");
    }
}

static char *BuildWhereClause()
{
    int i;
    int keyCount;
    char *buffer;
    int bufferLen;


    if( numKeys == 0 ) {
      return "";
    }
    if( boxColumn == NULL ) {
        DSUserError("keys specified but no boxcolumn argument");
    }

    bufferLen = numRanges*256+1 + 256;
    buffer = DSMalloc(bufferLen);
    buffer[0] = '\0';

    strncat(buffer," where overlap(", bufferLen);
    strncat(buffer, boxColumn, bufferLen);
    strncat(buffer,",DSRangeToBox('", bufferLen);
    keyCount = 0;
    for( i = 0; i < numRanges; i++ ) {
       if( rangeIsKey[i] ) {
```

```
            if( keyCount > 0 ) {
               strncat(buffer, ",", bufferLen);
            }
            strncat(buffer, rangeColumn[i], bufferLen);
            strncat(buffer, " ", bufferLen);
            strncat(buffer, rangeLow[i], bufferLen);
            strncat(buffer, " ", bufferLen);
            strncat(buffer, rangeHigh[i], bufferLen);
            keyCount++;
         }
      }
      strncat(buffer,"'))", bufferLen);
      return buffer;
}

static char * BuildQueryText() {
      int bufferLen;
      char *buffer;
      char *whereClause;

      whereClause = BuildWhereClause();
      bufferLen = strlen(whereClause) + numColumns*80 + 256;
      buffer = DSMalloc(bufferLen);
      buffer[0] = '\0';
      strncat(buffer, "select ", bufferLen);
      strncat(buffer, chipColumnName, bufferLen);
      strncat(buffer, " from ", bufferLen);
      strncat(buffer, tableName, bufferLen);
      strncat(buffer, whereClause, bufferLen);
      return buffer;
}

static void FetchChips()
{
   EXEC SQL BEGIN DECLARE SECTION;
       var binary "dschip" chipVar=NULL;
       char *queryText;
   EXEC SQL END DECLARE SECTION;
   DSChip *chip;
   int chipCnt;

   queryText = BuildQueryText();
   if( verbose ) {
      fprintf(stderr, "query was:\n%s\n", queryText);
   }

   /*
    * set the size of the fetch buffer to reduce the number of
roundtrips
    * to the server. The default is max(4096,one row).
    */
   FetBufSize = 32000;

   EXEC SQL whenever sqlerror CALL processESQLError;

   EXEC SQL connect to "noaa";
```

```
   EXEC SQL prepare coll_stmt from
          :queryText ;

   EXEC SQL declare noaacursor cursor for coll_stmt;

   EXEC SQL open noaacursor;

   for(chipCnt=0;;chipCnt++) {
      EXEC SQL fetch noaacursor into :chipVar ;

      if( strncmp( SQLSTATE, "00", 2) != 0 ) {
         break;
      }
      if( verbose ) {
         fprintf(stderr,"fetched chip\n");
      }
      chip = DSChipUnPack(chipVar);
      ProcessSingleChip(chip);
      DSChipFree(chip);
      ifx_var_dealloc(&chipVar);

   }
   if( verbose ) {
      fprintf(stderr,"about to close cursor\n");
   }
   EXEC SQL close noaacursor;
   EXEC SQL disconnect current;

   if( verbose ) {
      fprintf(stderr, "chip count was %d\n", chipCnt);
   }

}

static void SetVerbose() {
   verbose = 1;
}

static OptLstType MyOpts[] = {
    {  "o-", SetFileName },
    {  "table-", SetTableName },
    {  "boxcolumn-", SetBoxColumn},
    {  "columns-", SetColumn },
    {  "chip-", SetChipName },
    {  "key-", SetKey },
    {  "range-", SetRange },
    {  "verbose", SetVerbose },
    {  "\0", NULL }
};

static void printUsage(char *name) {
    fprintf(stderr,"Usage: %s arguments\n", name);
    fprintf(stderr,"where arguments include:\n");
    fprintf(stderr,  "\t-o output filename # default is stdout\n");
    fprintf(stderr,  "\t-table tablename\n");
    fprintf(stderr,  "\t-boxcolumn boxcolumn # for rtree indexing\n");
    fprintf(stderr,
```

```
          "\t-columns outputColumns # comma separated, may be
repeated\n");
    fprintf(stderr,  "\t-chip name_of_dschip_column\n");
    fprintf(stderr,  "\t-range 'columnname,lowValue,highValue'\n");
    fprintf(stderr,  "\t-key 'columnname,lowValue,highValue'\n");
    fprintf(stderr,  "Note: keys are special cases of range.\n");
}

int main(int argc, char *argv[]) {
    DSChip *chip;

    DSErrSource(argv[0]);

    if( argc == 1 ) {
       printUsage(argv[0]);
    }
    argc = GenParseOpt( argc, argv, MyOpts);
    if( argc != 1 ) {
       fprintf(stderr,"unrecognized arg: %s\n", argv[1]);
       printUsage(argv[0]);
    }
    if( tableName == NULL ) {
       DSUserError("missing -table argument");
    }
    if( numColumns == 0 ) {
       DSUserError("No -columns argument");
    }
    if( chipColumnName == NULL ) {
       DSUserError("No -chip argument");
    }
    if( outFileName != NULL ) {
        outfile = fopen(outFileName, "w");
        if( !outfile ) {
           DSUserError("Unable to open file '%s' for output");
        }
    }
    else {
        outfile = stdout;
    }
    FetchChips();
    if( outfile != stdout ) {
        fclose(outfile);
    }
    return 0;
}
```