1. Oracle Setup and Configuration

After ORACLE has been properly installed, it is necessary to build the ORACLE7 database tables. During the database configuration process, you will be asked to supply a system identifier (SID) and database name.

Use the following:

- **SID:** MISER
- **Database Name:** HISTORY

1.1 Creating Tablespaces

It is necessary to create the following tablespaces:

- **TEMP** - used while sorting and creating indexes
- **POINT** - stores points data

In addition, it is also necessary to create two additional tablespaces for the rollback segments.

NOTE The point definition file is named **`create_points_table.sql`**. See **Creating the Points Database Table** for details.

To bring up an SQL prompt, type the following at the OpenVMS system prompt:

- **`sqlplus system/manager`**

This will cause an SQL prompt to appear.

1.1.1 Creating the TEMP Tablespace

To create TEMP tablespace, type the following at the SQL prompt: (If it is necessary to enter a long string of text, hit RETURN where appropriate. ORACLE will automatically number the next line, and allow you to continue typing. Be sure to end the text with a semicolon character (;) to indicate that this is the end of the text string.)

```
SQL>   create tablespace TEMP  datafile
       2 [ORACLE7.DB_HISTORY]TEMP.DBS' size 20000K
       3 default storage ( initial 5000K NEXT 200K
       4 pctincrease 0 );
```

Now, to assign this temporary segment to the user **system**, type:

```
SQL> alter user system TEMPORARY tablespace TEMP;
```

1.1.2 Creating the POINT Tablespace

To create POINT tablespace, type at the prompt:

```
SQL>   create tablespace POINT  datafile
       2  '[ORACLE7.DB_HISTORY]POINTS.DBS'size 15000K
       3  default storage ( initial 1000K NEXT 1000K
       4  pctincrease 0 );
```

1.1.3 Creating Rollback Segments

To create additional rollback segments, type:

```
SQL>   create public rollback segment r0 tablespace system;
SQL>   alter rollback segment r0 ONLINE;
SQL>   create tablespace RBS1 datafile
       2  '[ORACLE7.DB_HISTORY]RBS1.DBS'size 10000K
       3  default storage ( initial 4000K NEXT 100K);
SQL>   create tablespace RBS2 datafile
       2  '[ORACLE7.DB_HISTORY]RBS2.DBS'size 10000K
       3  default storage ( initial 4000K NEXT 100K);
SQL>   create public rollback segment rbs1 tablespace rbs1;
SQL>   alter rollback segment rbs1 ONLINE;
SQL>   alter rollback segment r0 OFFLINE;
SQL>   create public rollback segment rbs2 tablespace rbs2;
SQL>   alter rollback segment rbs2 ONLINE;
```

The size setting for each datafile depends on the number of points in the database.

1.2 Creating the Points Database Table

It is necessary to create a "POINTS" table in which to store information about the Miser points. The table has two indices: one for acronym and one for the point number (**recnm**).

The table is divided into two main parts:

- The **fixed** part stores point definitions created through MISER utility DPT.
- The **variable** part stores current point data produced by Change OF State (COS) reports.

When a user defines a point in DPT, MISER creates a new entry in the ORACLE database and writes the information into the appropriate fixed and variable part of the ORACLE table.

- The *fixed* part of the table is updated only if the user later changes a point definition through DPT.
- The *variable* part of the table is updated each time a COS message changes the value or status of a point.

The structure of this table is defined in file **create_points_table.sql**. Because of its large size, it is easiest to enter this data into an ASCII text file, and then pass this file to the SQL editor.

See **MISER Technical Reference Manual**, Chapter 2, for details on each field in the MISER points database record structure.

To create the POINTs table, type the following at the SQL prompt:

```
SQL>   @[ path ]create_points_table.sql
```

1.2.1 create_points_table.sql

Note ORACLE interprets two sequential dashes (--) as a comment, and ignores the following text until a new line is entered.

```
Create table points (
-- fixed area
acronym   VARCHAR2(14) unique ,       -- character point acronym
recnm     number (5)  primary key ,   -- record number
relbin    VARCHAR2(14),               -- Related point
pntnam    VARCHAR2(12),               -- point name
area      VARCHAR2(12),               -- area point is in or near
bldng     VARCHAR2(12),               -- building point is in or near
unit      VARCHAR2(12),               -- unit point is associated with
tsknam    CHAR(8),                    -- related task name
-- binary point definition
on_code   VARCHAR2(8),
off_code  VARCHAR2(8),
```

```
mid_code  VARCHAR2(8),
strtv     VARCHAR2(8),                -- start command verb
stopv     VARCHAR2(8),                -- stop  command verb
-- analog point definition
code      VARCHAR2(8),                -- engineering unit print code
-- -------------------------------
node      CHAR(6),
ncc       number (5),
-- -------------------------------
rtu       number (5),
mux       number (5),
-- -------------------------------
inptn     number (5),                 -- input field address
otptn     number (5),                 -- output field address
-- -------------------------------
group_id  number (10),
-- -------------------------------
seg_num   number(2),                  -- segment nimber
rectyp    number(3),                  -- point type code
subtyp    number(3),                  -- input subtype code
osbtp     number(3),                  -- output subtype code
-- -------------------------------
tirm      number(3),                  -- time interval reporting interval size
aclvl     number(3),                  -- level required to access poin
cnlvl     number(3),                  -- level required to control point
relation  number(3),                  -- related point relationship type
-- -------------------------------
almprt    number (10),                -- bit map of Alarm Printer APn: (n=0,31)
-- -------------------------------
logprt    number (10),                -- loggin bit map Printer APn: (n=0,31)
-- -------------------------------
almdfn    number (10),                -- alarm handleing definition bits
pntdfn    number (10),                -- point definition bits
asdsn     number (10),                -- ASD slide number
-- -------------------------------
delay    number,                      -- alarm delay (seconds)
-- -------------------------------
dvspan    number,                     -- Analog span (Eng. Units/count)
dvbase    number,                     -- Analog base (Eng. Units)
filter    number,                     -- Analog filtering time constant
(seconds)
hilim1    number,                     -- Analog 1st level hi alarm limit (Eng.
Units)
lowlm1    number,                     -- Analog 1st level low alarm limit (Eng.
Units)
hilim2    number,                     -- Analog 2nd level hi alarm limit (Eng.
Units)
lowlm2    number,                     -- Analog 2nd level low alarm limit (Eng.
Units)
dbandh    number,                     -- Analog hi alarm deadband (Eng. Units)
dbandl    number,                     -- Analog low alarm deadband (Eng. Units)
ratlim    number,                     -- Analog rate of change limit (Eng.
Units/hour)
spllo     number,                     -- Analog output lo limit (Eng. Units)
splhi     number,                     -- Analog output hi limit (Eng. Units)
ptkw      number,                     -- Binary point demand (KW)
-- -------------------------------
tolrnc    number (5),                 -- Analog COS reporting tolerance (% *
100)
shlcc     number (5),                 -- Analog sensor high limits converter
counts
-- -------------------------------
```

```
sllcc     number (5),                 -- Analog sensor low limits converter
counts
fmtcd     number (5),                 -- Analog display format code (# of
dec.places)
-- --------------------------------
vfdly     number (5),                 -- Binary verification delay (seconds)
minon     number (5),                 -- Binary minimum on time (seconds)
-- --------------------------------
minoff    number (5),                 -- Binary minimum off time (seconds)
bonst     number (5),                 -- Binary point on status
-- --------------------------------
boffst    number (5),                 -- Binary point off status
bintst    number (5),                 -- Binary point intermediate status
-- --------------------------------
Node_index number (5),                -- unit index of Node in SYSCOM
NCC_index  number (5),                -- unit index of NCC in SYSCOM
-- --------------------------------
RTU_index  number (5),                -- unit index of RTU in SYSCOM
MUX_index  number (5),                -- unit index of MUX in SYSCOM
-- --------------------------------
rel_num   number (5),                 -- related point record number
msgnum    number (5),                 -- related message number
-- --------------------------------
tskpnt    number (5),                 -- history
alarm_pri_level number (5),           -- alarm priority level
rtuport   number (5),
--     variable area
pntsts    number (10),                -- point status
almsts    number (10),                -- alarm status

-- analog point
anarv     number ,                    -- analog current point input value (Eng
u)
lanarv    number,                     -- last value input
spval     number,                     -- real output
cosct     number (10),                -- change of state count
-- binary point
blscm     number (5),                 -- binary last command (output value)
cntrl     number (5),                 -- control owneship bits
bcrvl     number (5),                 -- binary current input value
blsvl     number (5),                 -- binary last input
anaval    number (5),                 -- Analog current unscaled reading
-- time stamps
tr_time   DATE,                       -- most recent tr.
tscur     DATE,                       -- most recent input value
alm_time  DATE,                       -- alarm
tsout     DATE,                       -- output value
lston     DATE,                       -- last ON COS time
lstoff    DATE,                       -- last OFF
text_value VARCHAR2(60) )             -- text point
tablespace POINT;
```

1.3 Creating an Alarm Status Table

To create the alarm_status table, enter the following at the SQL prompt:

```
SQL>   create table alarm_status (
       2 code VARCHAR2(12), value number (3),
       3 rectyp number (2));
```

To display the contents of the alarm_status table, type the following:

```
SQL>select * from alarm_status;
```

The following table will display.

```
CODE              VALUE     RECTYP
------------ ---------- ----------
LO                    1          1
HI                    3          1
LO_LO                 4          1
HI_HI                12          1
RATE                 16          1
LIMITS               32          1
MULFUNTION          128          1
MULFUNTION          128          2
UNCOMMANDED          64          2
VERIFICATION        256          2
OFF                   1          2
ON                    2          2
INTERMEDIATE          3          2
UNDEFINE             32          2
```

NOTE: RECTYP is 1 for analog point, and 2 for binary points.

1.4 Creating a History Table

It is necessary to define a “HST” table for the History data file. This HST table is used by HYPROC as a prototype in creating the history tables. HYPROC creates a separate table for each new day, named AYYMMDD, where

- A - the initial (mandatory) alpha-character
- YY - the year (00 - 99)
- MM - the month (01 - 12)
- DD - the day (01 - 31)

HYPROC creates an index for each table. The index (IND_AYYMMDD) for each table is based on the **point_number** and **time** parameters. The table structure is shown below.

```
Create table hst (
point_number number (6),
time         date,
value        number,
alarm        number (3),
quality      number,
type         number (3),
interval     number (3),
ms           number  );
```

To create a HST table, type the following at the prompt:

```
SQL> @SYS$SYSDEVICE:[ORACLE7.DBA]CREATE_HST_TABLE.SQL
```

1.5 Conversion Issues

To perform SQL queries on the MISER points database and the daily MISER history files, it is first necessary to convert each to a format suitable for this process.

- Instructions on how to convert MISER file points.dat (the MISER points database) to an SQL-compatible format are covered in the **Converting MISER Point Definitions** section below.
- Instructions on how to convert MISER daily history files to an SQL-compatible format are covered in the **Converting MISER History Files** section below.

1.5.1 Converting MISER Point Definitions

MISER point definitions are stored in MISER file points.dat. Utility program PCVT is used to convert the points database into a format suitable for SQL queries. The following MISER files are necessary for this procedure:

- **BUILD.COM**
- **PCVT.FOR**
- **VALFMT.FOR**
- **COPY.MAR**

To build PCVT and to use it to convert the MISER points database, perform the following steps:

1. Enter the following command at the VMS system prompt:

   ```
   @build.com
   ```

   This action will create (i.e., build) PCVT executable file, PCVT.EXE.

2. Copy **points.dat** into the local directory and run PCVT.EXE.

   ```
   $ copy [miser.data]points.dat []*.*
   ```

   ```
   $ run pcvt
   ```

   This action will create a **convert.sql** file in the local directory.

3. At the system prompt, enter the following commands complete the conversion process:

   ```
   $ sqlplus system/manager
   ```

   ```
   SQL? @convert.sql
   ```

   This action will place a copy of points.dat suitable for performing SQL queries in the local directory.

1.5.2 Converting MISER History Files

**WARNING If a table already exists for the date specified,
it will be automatically deleted and a new (blank) table created in its place.**

The MISER CNVN utility program enables the conversion of a MISER history file into an ORACLE table. It performs this task on a single day's history at a time.

The CNVN program can be specified as a logical command as shown below:

```
CNVN == "$MNET$EXE:CNVN"
```

1.5.2.1 Usage

To convert a day's history, enter the following command at the VMS system prompt:

```
$ CNVN date
```

where *date* is the date for which history conversion is desired and is specified in a DD-MMM-YEAR format. The resulting output file will be named MnetHST:Ayymmdd.sql where the letter A does not change and yymmdd specifies the day in YY-MM-DD format.

An example conversion procedure for September 2, 1997 is shown below.

To convert the whole history file from 2-Sept-1997, invoke the CNVN conversion utility from a VMS system prompt:

```
$ CNVN 2-sep-1997
```

This will convert file Mnet$HST:970902A.HST into a new output file called Mnet$HST:A970902.sql,

An example output file is shown below.

```
DROP TABLE A970902 ;
CREATE TABLE A970902  AS SELECT * from hst;
INSERT into A970902  values (    7, to_date('02-09-97 00:00:00 ' ,
                'dd-mm-yy hh24:mi:ss'),    2425032.000, 0, 0, 1, 0,      0);
INSERT into A970902  values (    7, to_date('02-09-97 00:00:05 ' ,
                'dd-mm-yy hh24:mi:ss'),    2425131.000, 0, 0, 2, 0,      6);
INSERT into A970902  values (    7, to_date('02-09-97 00:01:01 ' ,
                'dd-mm-yy hh24:mi:ss'),    2425122.000, 0, 0, 2, 0,    252);
INSERT into A970902  values (  153, to_date('02-09-97 00:00:00 ' ,
                'dd-mm-yy hh24:mi:ss'),          2.000, 0, 0, 1, 0,      0);
INSERT into A970902  values (  157, to_date('02-09-97 00:00:00 ' ,
                'dd-mm-yy hh24:mi:ss'),          1.000, 0, 0, 1, 0,      0);
INSERT into A970902  values (  196, to_date('02-09-97 00:00:00 ' ,
                'dd-mm-yy hh24:mi:ss'),          0.000, 0, 4, 1, 0,      0);
commit;
```

This output file *must be* executed on an ORACLE server. This can be done from VMS node or from NT ORACLE server.

1.5.2.2 From a VMS Node

To perform the history conversion from a VMS node, perform the following steps:

1. Log on to the node as ORACLE7.
2. At the VMS prompt, type the following command:

```
sqlplus system/manager@cds
```

3. At the subsequent sqlplus> prompt, type:

```
sqlplus> @mnet$hst:A970902.sql
```

1.5.2.3 From an NT ORACLE Server

To perform the history conversion on the NT workstation, perform the following steps:

1. Transfer output file Mnet$hst:A970902.sql via FTP (File Transfer Protocol) from the MISER host to the NT workstation. (Remember to use binary mode for history file transfer operations.)
2. In an MS-DOS window, enter the following command at the DOS prompt:

```
C:\ sqlplus system/manager
```

3. At the subsequent sql> prompt, type:

```
sql> @ A970902.sql
```

   The records specified in A970902.sql file will be inserted in the table just created.

NOTE The time required to insert the history records for an entire day can be quite lengthy depending on the number of records involved for that particular day..

1.6 ORACLE Database Requests

Three examples are provided to illustrate the SQL statements necessary to create subsets of points listings in the ORACLE database

- **SQL Example 1** creates a point listing using an acronym and wildcard.
- **SQL Example 2** creates a subset of the point listing in Example 1 which includes only those points also in alarm.
- **SQL Example 3** a subset of the point listing in Example 2 which includes only those points in HI-HI alarm state.

1.6.1 SQL Example 1

Example 1 creates a listing of all points in the ORACLE database with the acronym ORACLE* .

```
SQL>   select  ACRONYM, RECNM POINT_NUMBER,
       2  ANARV "CURRENT VALUE", a.code ALARM
       3  from points p , alarm_status a
       4  where ACRONYM like 'ORACLE%' AND mod (ALMSTS,
       5  256)=value(+) and p.rectyp = a.rectyp(+);
```

The following table will display.

```
ACRONYM        POINT_NUMBER CURRENT VALUE ALARM
-------------- ------------ ------------- ------------
ORACLE25            162          44
ORACLE35            163           0
ORACLE61            158           3
ORACLE6             161          55
ORACLE55            164           0
ORACLE7             159           0
ORACLE3             155           2
ORACLE5             160          -150      LO
ORACLE1             153           110      HI
ORACLE4             156          -250      LO_LO
ORACLE2             154           250      HI_HI
```

1.6.2 SQL Example 2

Example 2 creates listing of points in the ORACLE database with the acronym ORACLE* *and* which are also in an alarm state.

```
SQL>   select ACRONYM, RECNM "POINT NUMBER",
       2  ANARV "CURRENT VALUE", a.code "ALARM"
       3  from points p , alarm_status a
       4  where ACRONYM like 'ORACLE%' AND mod (ALMSTS,
       5  256)=value and p.rectyp = a.rectyp;
```

The following table will display.

```
ACRONYM        POINT NUMBER CURRENT VALUE ALARM
-------------- ------------ ------------- ------------
ORACLE5             160          -150       LO
ORACLE1             153           110       HI
ORACLE4             156          -250       LO_LO
ORACLE2             154           250       HI_HI
```

1.6.3 SQL Example 3

Example 3 create a listing of all points in the ORACLE database with acronym ORACLE* AND which are also in the HI_HI alarm state.

```
SQL>   select  ACRONYM, RECNM "POINT NUMBER",
       2  ANARV "CURRENT VALUE"
       3  from points p , alarm_status a
       4  where ACRONYM like 'ORACLE%' AND mod (ALMSTS,
       5 256)=value and a.code like 'HI_HI'
       6 and p.rectyp = a.rectyp;
```

The following table will display.

```
ACRONYM        POINT NUMBER CURRENT VALUE
-------------- ------------ -------------
ORACLE2             154          250
```

1.7 Extracting Data from the History Database

Data is extracted from the ORACLE database using four utility procedures:

- **HAVG** - Returns the average value for a point over a given time interval.
- **HMAX** - Returns the maximum value (and the time when the maximum value occurred) for a point over a given time interval.
- **HMIN** - Returns the minimum value for a point (and the time when the minimum value occurred) over a given time interval.
- **HEND** - Returns the value of a point at the end of a given time interval.

Each procedure returns data for one point over a given time interval. This interval can range from hours to years. Using these stored procedures, it is possible to retrieve the maximum, minimum and average values for a day, a month, or a year.

Each of the four defined procedures have a similar interface. For example, for HMAX:

```
HMAX (  point_number,  -- point number from POINTS.dat
        start_time,    -- begining of interval in DD-MON-YY HH:MI:SS
        stop_time,     -- end of interval in DD-MON-YY HH:MI:SS,
        start_value,   -- value at the begining of interval (opt.),
        start_quality, -- quality at the begining of interval (opt.),
        start_alarm,   -- alarm at the begining of interval (opt.),

        result_value,  -- result value - maximum value for the point,
        result_time,   -- time when max occure (DD-MON-YY HH:MI:SS),
        result_quality,-- quality of the point when max occure,
        result_alarm)  -- alarm of the point when max occure.
```

In addition to returning the item of interest (average, maximum, minimum or end-of-interval value) and the time at which it occurred, each procedure also returns the end-of-interval value as the **start_value** parameter. In a similar fashion, the end-of-interval data quality and alarm status are returned as the **start_quality** and **start_alarm** values. These end-of-interval values are used as initial conditions for an iteration during a loop sequence.

NOTE If the data quality at the beginning of the time interval is unknown, the **start_quality** parameter must be set to H_NOT.

1.7.1 History Data Retrieval

To retrieve history data from the ORACLE database for use by MISER programs like PRP, ANT, and RUPROC, a number of steps must be performed.

1. The program must first call MISER routine Mnet$HISTORY.

2. Mnet$HISTORY then sends a message to MISER routine Mnet$HSTRY, and HSTRY, in turn, resends the message to HST00x.
3. HST00x receives this message and calls one of the stored procedure (HAVG, HMAX, or HMIN...) to process the request.
4. The retrieved data is returned to the initial calling program.

1.8 Modifications Made to MISER

Necessary changes made to MISER to facilitate the interaction between it and the ORACLE program are listed in the table below.

New Routines:	Mnet$LIB: CALL_STORED_PROC.PC
Modified Routines:	Mnet$HSTRY, Mnet$HISTORY
Modified Programs:	HYPROC , HSTLOOK, UPDATE, DSKIO
New Stored Procedures:	[oracle7.dba.proc_asc]HAVQ_STORE_PROC.SQL [oracle7.dba.proc_asc]HMAX_STORE_PROC.SQL [oracle7.dba.proc_asc]HMIN_STORE_PROC.SQL [oracle7.dba.proc_asc]HEND_STORE_PROC.SQL

HYPROC: MISER Program HYPROC has been modified to enable it to write history data to both the local (VMS) disk and to the remote ORACLE (NT workstation) server.

- HPROC writes to the local (VMS) disk as always.
- HYPROC writes to the remote ORACLE server *from the online node* **only** if logical **ORA_SID** has been defined as **MISER**.

HSTLOOK: MISER program HSTLOOK (used in accessing history data files) has been modified so that it uses logical HISTORY_LOCATION to determine the location of the history files.

- If logical HISTORY_LOCATION has been set to LOCAL, the history files (in VMS format) are located on the local hard drive.
- If logical HISTORY_LOCATION has been set to REMOTE, the history files (in ORACLE format) are located on the remote ORACLE server.

UPDATE: MISER program UPDATE periodically sends all changes that have been made to the points database to the remote ORACLE server.

DSKIO: MISER program DSKIO forwards every change that is made to a MISER point to the remote ORACLE server.

1.9 Network Failure Recovery

If the remote ORACLE server goes down, the MISER host will try to keep the connection open for 10 minutes. The MISER node will try to reconnect with the remote ORACLE server every five minutes until the connection is reestablished.

A new history table will be created on the ORACLE server if the table does not exist when the connection is reestablished.

1.10 Windows NT Configuration

The following software must be installed on the Windows NT server and configured correctly for the ORACLE database to be accessible to the MISER host.

Microsoft BackOffice Server Version 4.0

Microsoft Windows NT Server 4.0, Service Pack 3
Windows NT 4.0 Option Pack
Microsoft Internet Explorer 4.01
Microsoft SQL Server 6.5
Microsoft Exchange Server 5.5
Microsoft SNA Server 4.0
Microsoft Systems Management Server 1.2
Microsoft Proxy Server 2.0
BackOffice Server Intranet Starter Site
Seagate Crystal Info 5.0

ORACLE

Oracle7 Workgroup Server, Release 7.3 for Windows NT
Developer/2000, Release 2.0 for Windows NT and Windows 95

Intellution

FIX 6.12