MARINE CORPS HOUSING AUTOMATED SYSTEM, MCHAS Administrator's Workshop

MCHAS Version 2.5a
September 2002

MCHAS System Administrator Workshop

Agenda

The Database

1. Installing the Oracle Software Asset Manager
2. Installing Personal Oracle
3. Installing Developer Runtime
4. Installing SQR Work Bench
5. Installing MCHAS
6. Creating a MCHAS Backup
7. Recovering MCHAS from an Oracle Backup
8. Intro to SQL-Plus
    - Connecting to the Database
    - Setting the Environment
    - Spool Files
9. Overview of an Oracle Database
    - SID
    - Table Spaces
    - Extents
    - Init(sid).ora file
    - Rollback Segments
    - Altering System Parameters
10. Intro to DML
    - Select Statement
    - Insert Statement
    - Update Statement
    - Delete Statement
11. Intro to DDL
    - Database Objects
        - Users
        - Tables
        - Views

- Indexes
- Sequencers
- Packages and Procedures
- Synonyms
- Triggers
- Constraints

Creating Database Objects

- User
- Tables
- Views
- Indexes
- Synonyms
- Triggers

Altering Database Objects

- Users
- Tables

MCHAS System Administration

1. MCHAS System Admin Module

   Modules and Components

   Installation Info

   Security Setup

   Adding SQR Reports

   Setting User Passwords

   Data Cleanup

   Value Tables

   - Pay Grades
   - Zip Codes
   - Value Lists

   CRLOG

2. MCHAS Data Dictionary

   ALL_TAB_COLUMNS

   SQR DATADICT Report

3. MCHAS Table Relationships

   Sponsor Tables

   Housing Unit Tables

   Wait List Tables

SQR Reports

1. Introduction to SQR report writing
    - Connecting to a Database
    - General SQR Sections
        - Setup Section
        - Report Section
        - Header
        - Footer
    - Creating a Tabular Report
        - Format a Tabular Report
        - Adding a Header and Footer
        - Adding Totals
    - Join Tables
        - 1 to 1
        - 1 to Many
    - SQR Functions
    - Complex Queries
    - Creating a Student Defined Report

NOTES:

NOTES:

The Database

Installation of Tools

Because of certain anomalies and incompatibilities caused by the Installers on the Personal Oracle8 and Oracle Developer CD's, ORACLE® has recommended the use of their Installer 3.3.1.2.4 for the installation of a local database. These instructions cover the installation of this Installer and the subsequent Oracle 8 and Developer Runtime files.

Installing the Software Asset Manager

1. From the Start Menu, select Run (Figure 1).

Figure 1 - Windows Start Menu

2. Select Browse (Figure 2).

Figure 2 - Run Dialog

3. Navigate to the location of the Universal Installer (Figure 3).

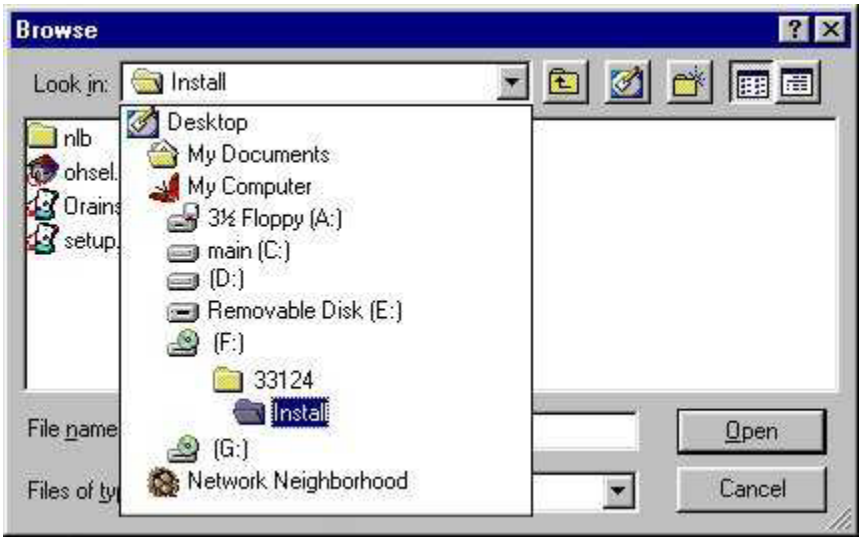


Figure 3 - Browse Window, navigation to Installer

4. Select Orainst.exe (Figure 4).

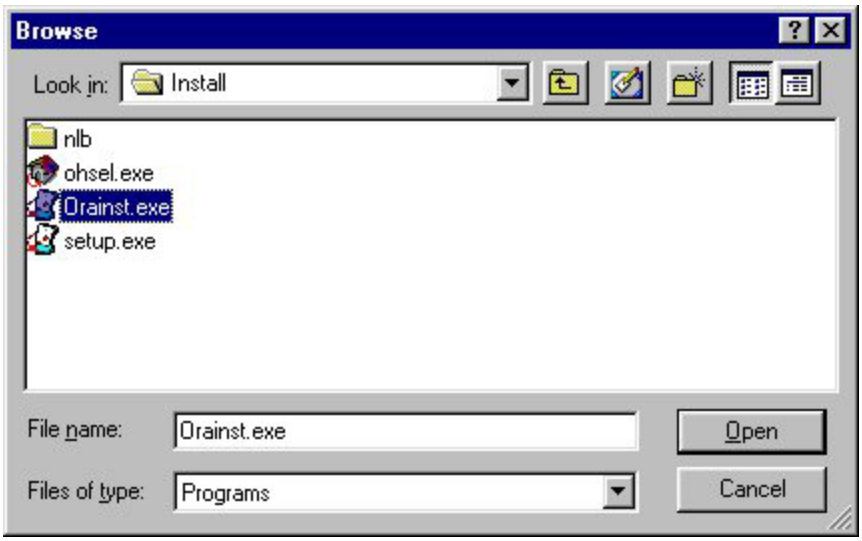


Figure 4 - The Universal Installer executable

5. Select OK (Figure 5) to begin the installation.

Figure 5 - Run again with the selected executable

Figure 6 - Loading the Installer

6. Enter the Company Name and Click OK (Figure 7).

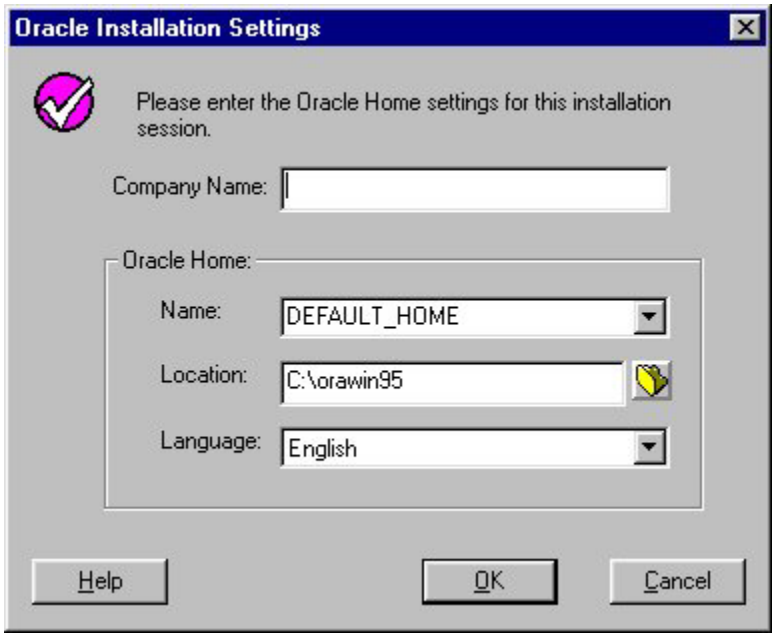


Figure 7 - Installation Settings

7. From the Software Asset Manager (Figure 8), Select Oracle Installer 3.3.1.2.4. The Install button between the two panes will become accessible, click it.

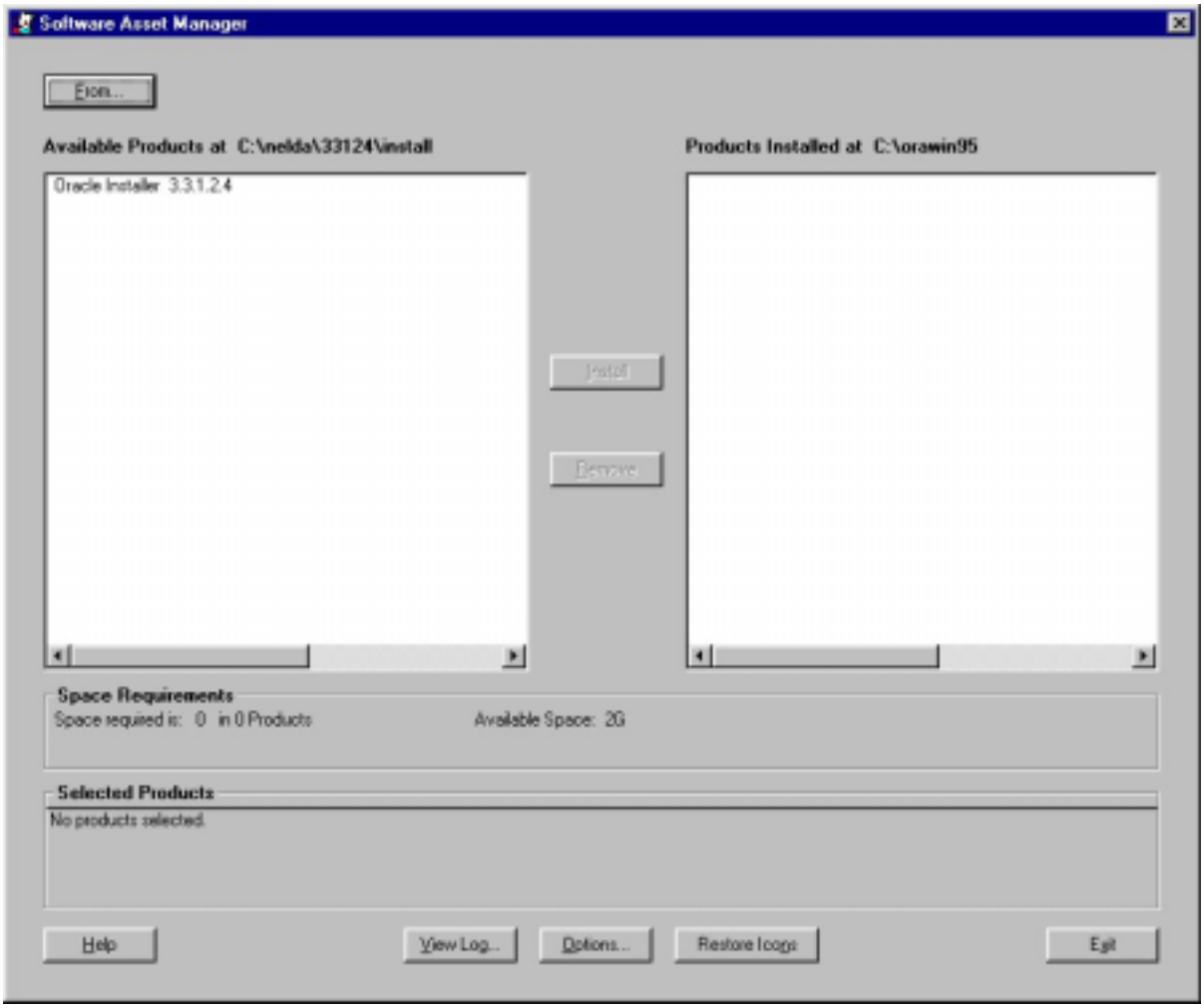


Figure 8 - Software Asset Manager

Figure 9 - Installation progress

8. You will now return to the Software Asset Manager showing the Installer listed under "Products Installed at…". (Figure 10)

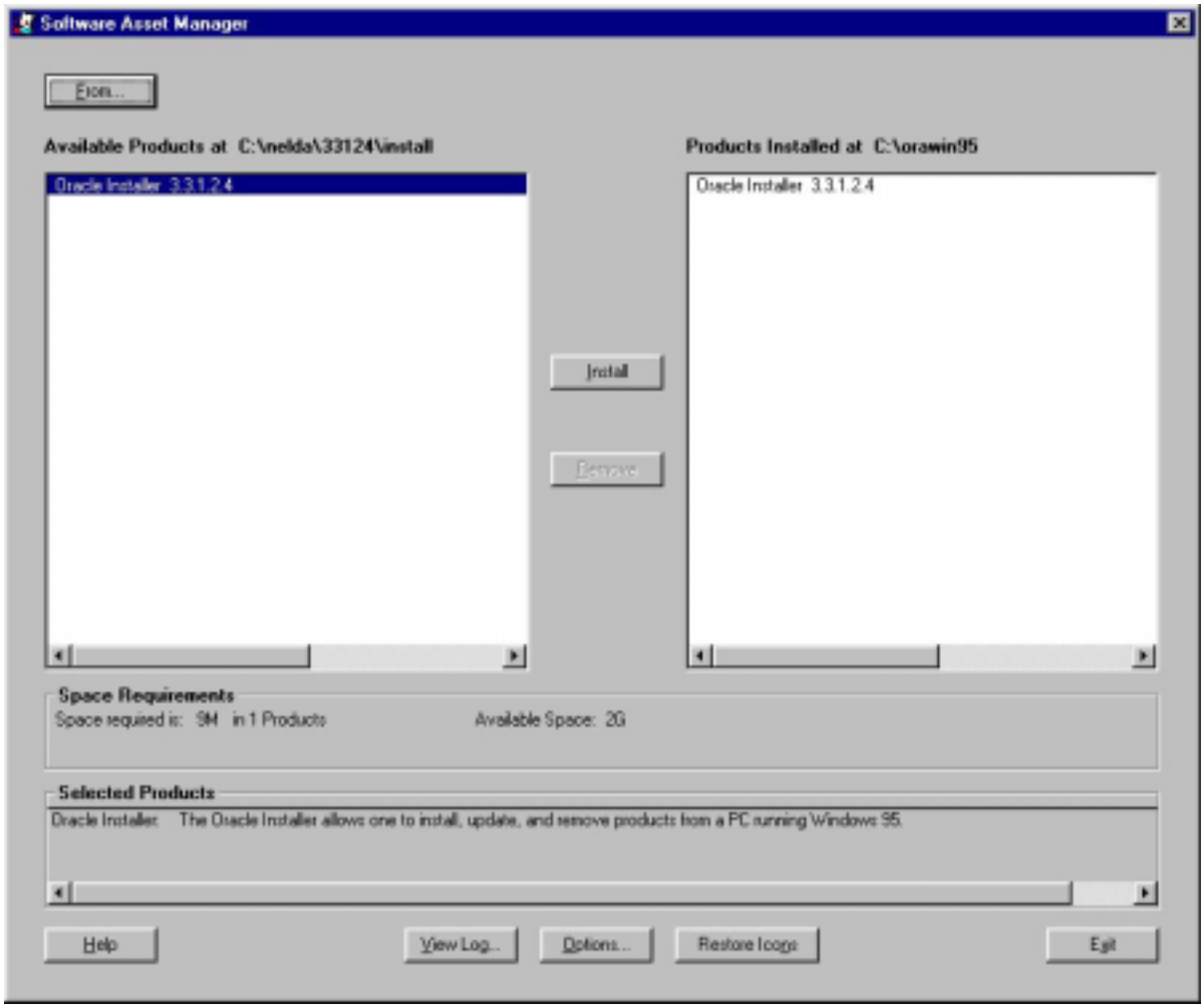


Figure 10 - Software Asset Manager after install

9. You may exit the Software Asset Manager by clicking the Exit button in the bottom right corner of the window or continue with the next installation.

Installing Personal Oracle 8

1. If you closed the Software Asset Manager from the previous session, Open it again.
2. From the Start Menu, Select Programs, Oracle for Windows 95 or Winnt, whichever operating system you have, and select Oracle Installer.
3. The Software Asset Manager will open (Figure 10). Click the From button in the upper left of the window.
4. Insert the Personal Oracle CD in the CD drive and navigate to the installation PRD file for Personal Oracle 8 (Figure 11). This path and file will be different for Windows 95/98 and Windows NT. Be sure to select the path for YOUR operating system.

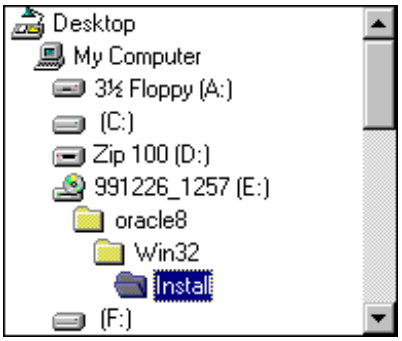


Figure 11 – Path to the installation PRD file

5. When the Software Asset Manager again opens, the products on the CD will be listed in the left pane. Expand the product lists by clicking on the + in front of the products that have one to expand the view.

```
Application Developer (Complete) Package Products
Runtime (Database Only) Package Products
-{All Products}
    Assistant Common Files  1.0.1.0.0
    Java(TM) Runtime Environment  1.1.1.0.0
    Oracle Call Interface  8.0.4.0.0
    Oracle Data Migration Assistant  8.0.4.0.0
    Oracle Documentation  8.0.4.0.0
    Oracle Installer  3.3.0.1.3
    Oracle Migration Assistant for Microsoft Access  8.0.4.0.0
-   Oracle Net8 Products  8.0.4.0.2c
        Oracle Net8 Add-on  8.0.4.0.0
        Oracle Net8 Assistant  8.0.4.0.3
        Oracle Net8 Client  8.0.4.0.2c
-       Oracle Protocol Adapters  8.0.4.0.2c
            Oracle Named Pipes Protocol Adapter  8.0.4.0.0
            Oracle SPX Protocol Adapter  8.0.4.0.0
            Oracle TCP/IP Protocol Adapter  8.0.4.0.2c
-   Oracle Networking Products  2.3.4.0.2
-       Oracle Protocol Adapters  2.3.4.0.0
            Oracle Named Pipes Protocol Adapter  2.3.4.0.0
            Oracle SPX Protocol Adapter  2.3.4.0.0
            Oracle TCP/IP Protocol Adapter  2.3.4.0.0
        SQL*Net Add-on  2.3.4.0.2
        SQL*Net Client  2.3.4.0.2
    Oracle Objects for OLE  2.2.2.2.0
    Oracle Patch Assistant  1.0.0.0.1
    Oracle7 Utilities  7.3.4.0.0
    Oracle8 JDBC Drivers  8.0.4.0.0
-   Oracle8 Navigator  8.0.4.0.0
        Online Help  8.0.4.0.0
    Oracle8 ODBC Driver  8.0.4.0.0
-   Oracle8 Personal Edition  8.0.4.0.0
        Oracle8 Database  8.0.4.0.0
        Oracle8 Utilities  8.0.4.0.1
    Oracle8 Personal Edition Release Notes  8.0.4.0.0
    Required Support Files  7.3.4.1.1
    Required Support Files  8.0.4.0.4
    SQL*Net Add-On(patch)  2.2.2.1.1
    SQL*Plus  8.0.4.0.0
```

Figure 12 – Oracle 8 Expanded List of Products

6. From the list select the following products by holding down the Ctrl key and selecting the product from the list.

   A. Oracle Net8 Assistant 8.0.4.0.3

B. Oracle Net8 Client 8.0.4.0.2c
C. Oracle TCP/IP Protocol Adapter 8.0.4.0.2c
D. Oracle8 Navigator 8.0.4.0.0
E. Oracle8 ODBC Driver 8.0.4.0.0
F. Oracle8 Personal Edition 8.0.4.0.0
G. SQL *Plus 8.0.4.0.0

7. Again the Install button will become accessible, click it to begin installing products.
8. Keep the default Standard (40M disk space) for the starter database (Figure 13), click OK.

Figure 13 - Starter Database Options

9. Click Yes to accept the default language (Figure 14).

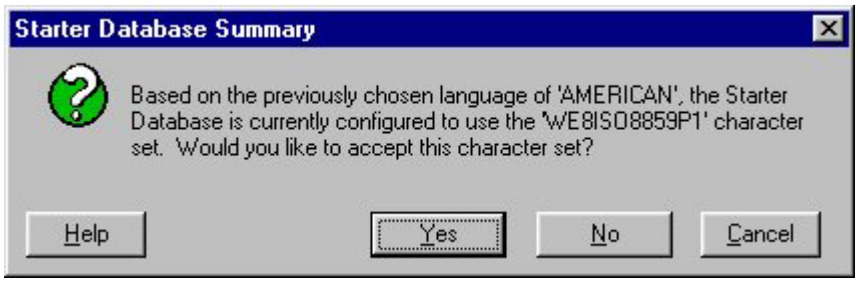


Figure 14 - Starter Database Summary

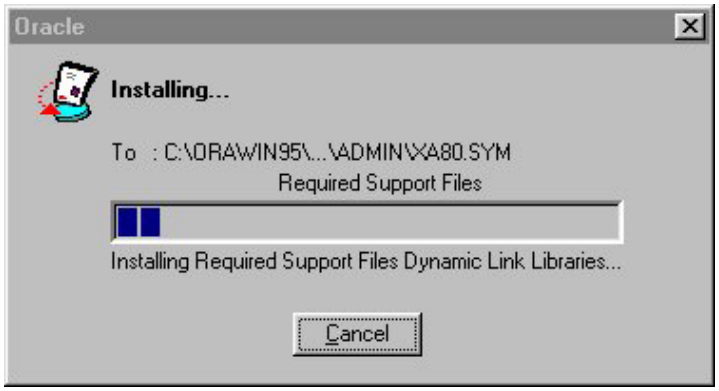


Figure 15 - Installation progress

10. The Software Asset Manager now shows the Oracle8 products installed. Note that certain files have been installed (Required Support Files 8.0.4.0.0) even though you did not chose them (Figure 16).

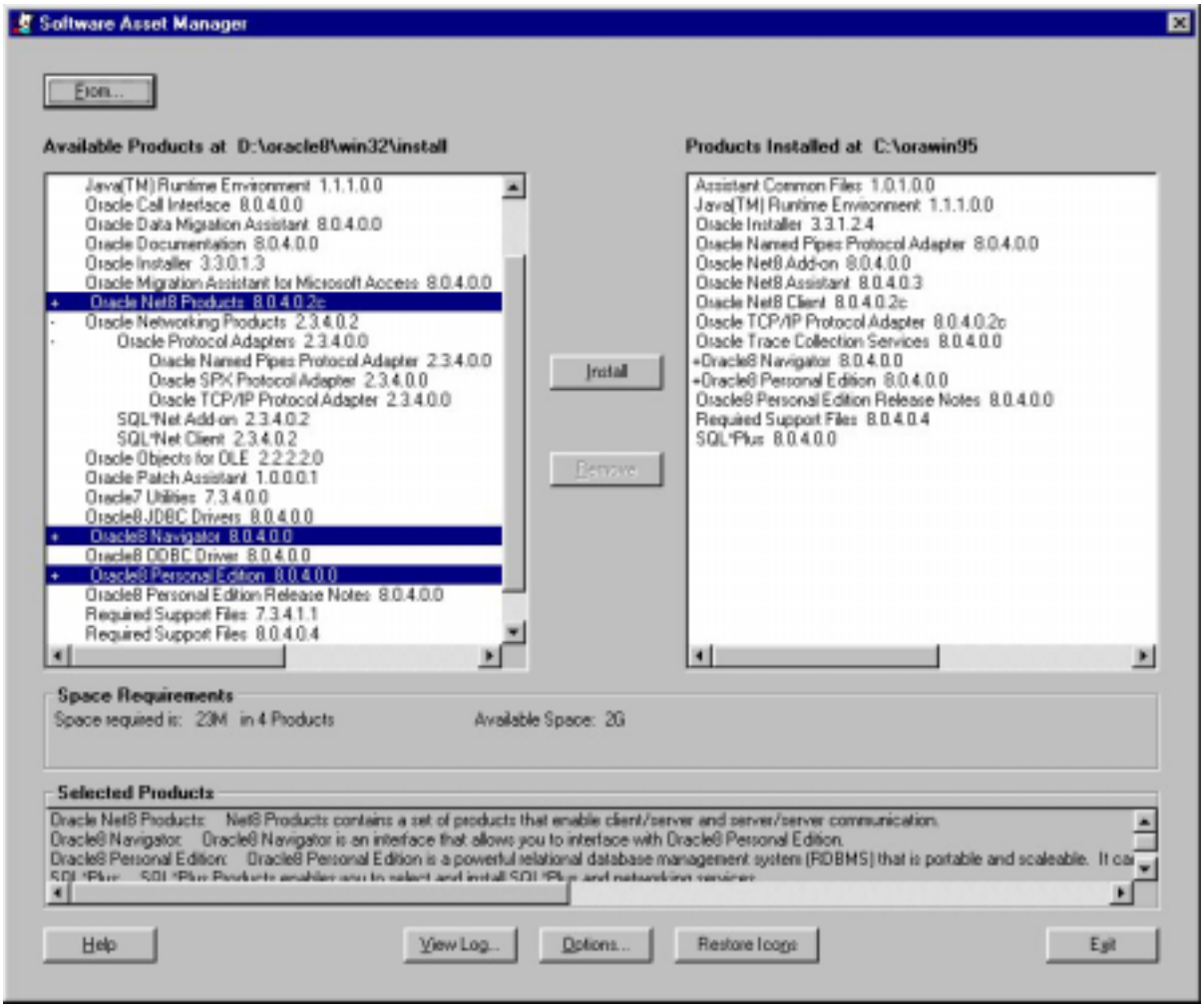


Figure 16 - Software Asset Manager with Oracle 8 installed

11. You may exit the Software Asset Manager by clicking the Exit button or keep it open for the next installation.

Install the Developer Runtime

1. If you closed the Software Asset Manager from the previous session, Open it again.
2. From the Start Menu, Select Programs, Oracle for Windows 95 or Winnt, whichever operating system you have, and select Oracle Installer.
3. The Software Asset Manager will open (Figure 16). Click the **From** button in the upper left of the window.
4. Insert the Developer Forms CD in the CD drive and navigate to the installation PRD file for Oracle Developer Forms (Figure 17). This path and file will be different for Windows 95/98 and Windows NT. Be sure to select the path for YOUR operating system.

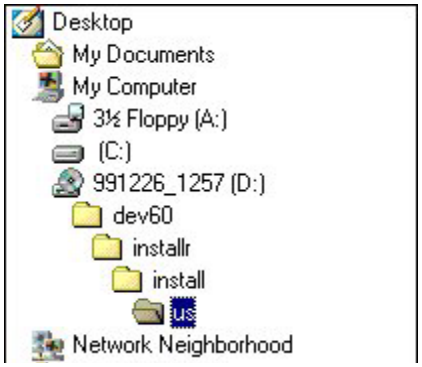


Figure 17 - Path for Developer Installation PRD file

5. The Software Asset Manager will open with the Developer Products listed in the left pane (Figure 18). Click on the + in front of a product to see more details about the products to be installed.

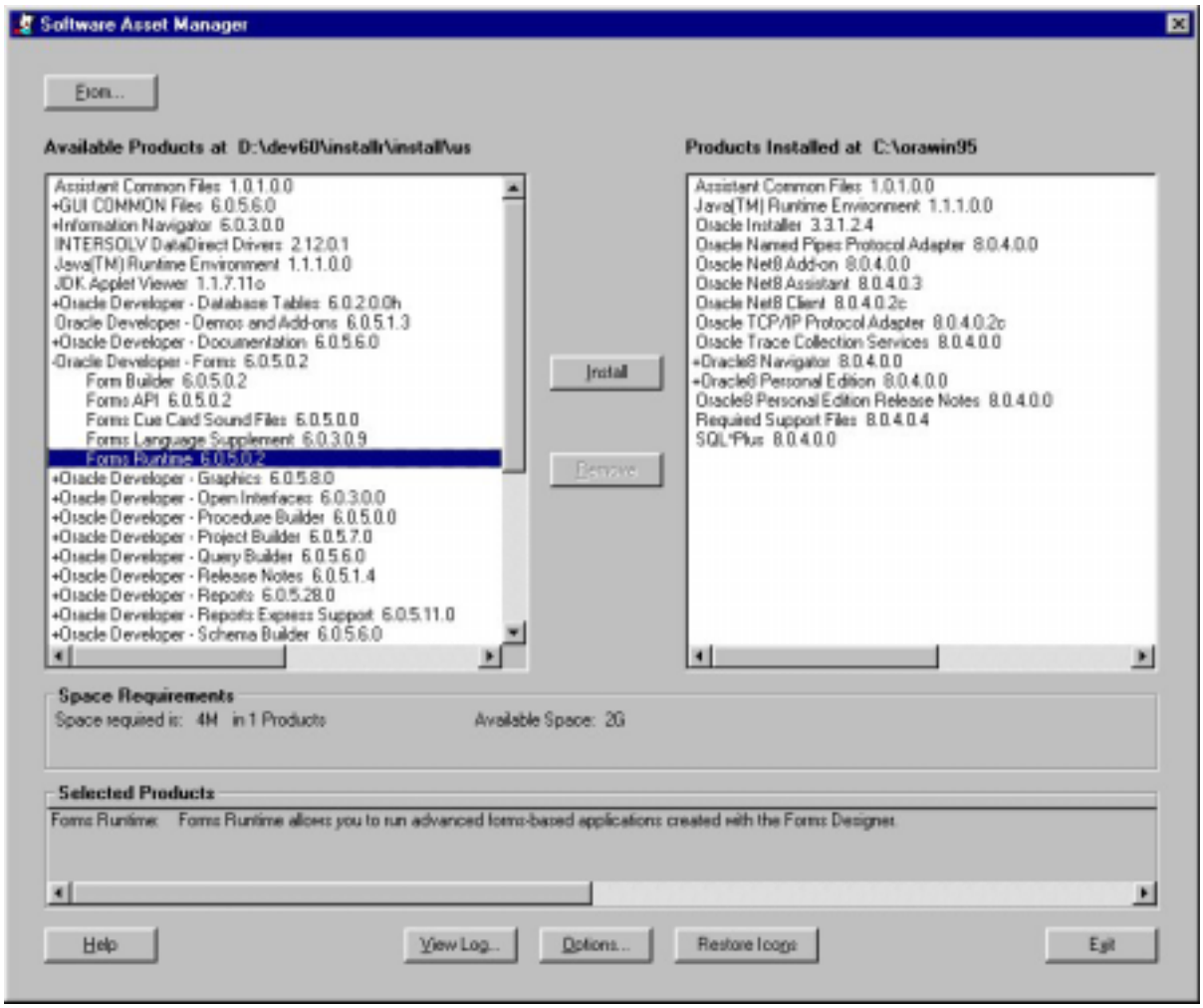


Figure 18 - Software Asset Manager

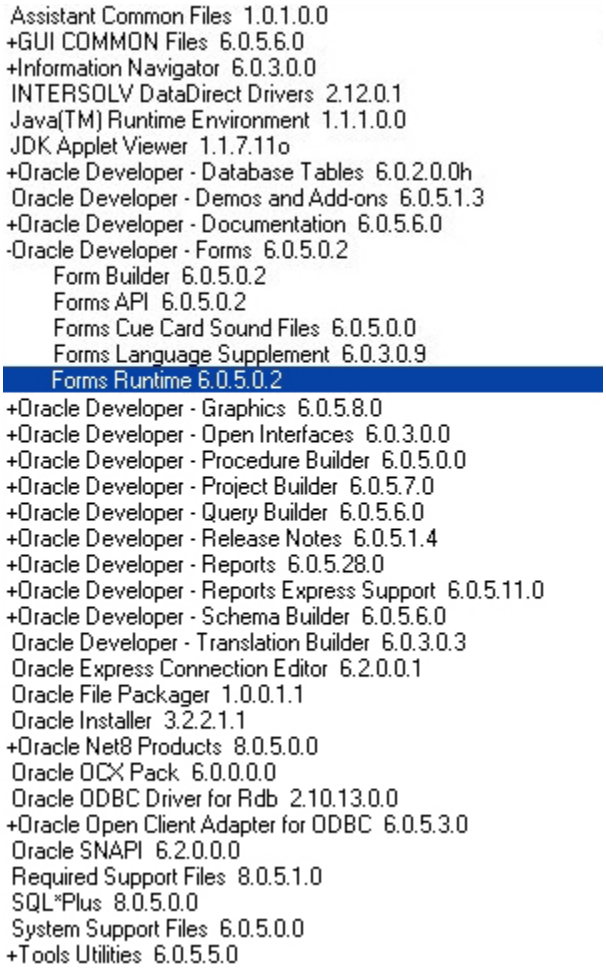


Figure 19 - Forms Product List

6. For a minimum install to run MCHAS on a local machine, select Forms Runtime 6.0.5.0.2. The necessary support files will automatically be installed.
7. Select the default folder for Forms60 and acknowledge the System Support Files message by clicking OK (Figure 20 and Figure 21).

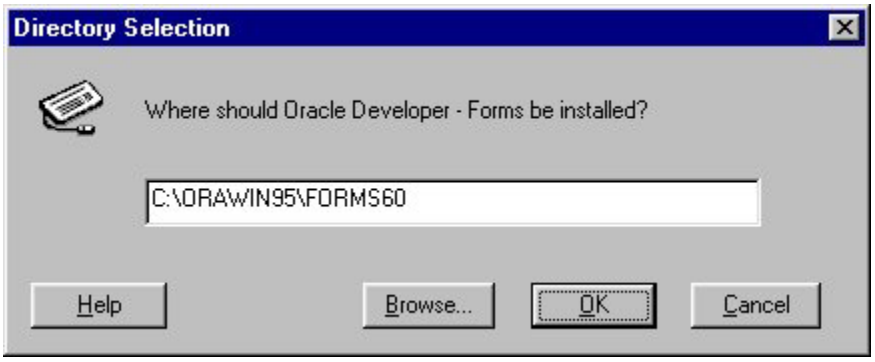


Figure 20 - Directory Selection for Forms

Figure 21- System Support Files dialog

8. Continue the installation by clicking Yes when you receive the Database Conflict dialog (Figure 22).

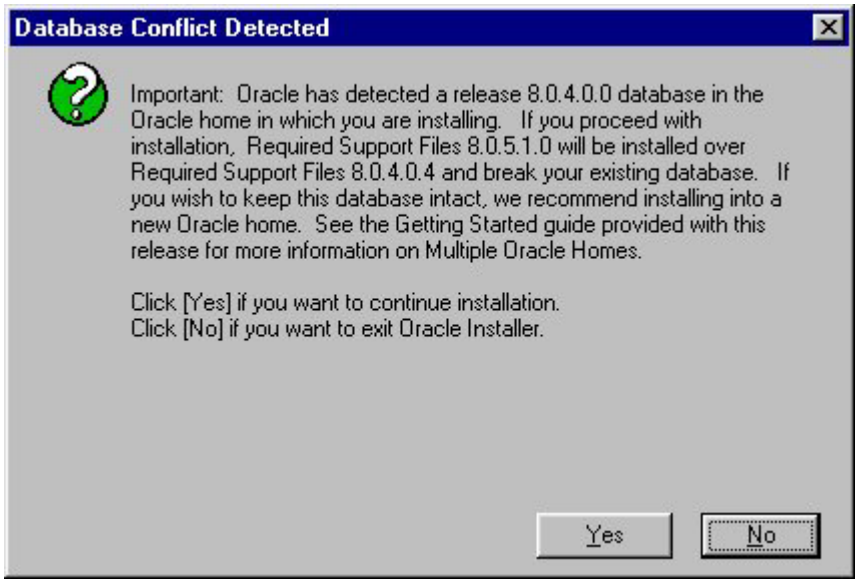


Figure 22 - Conflict dialog

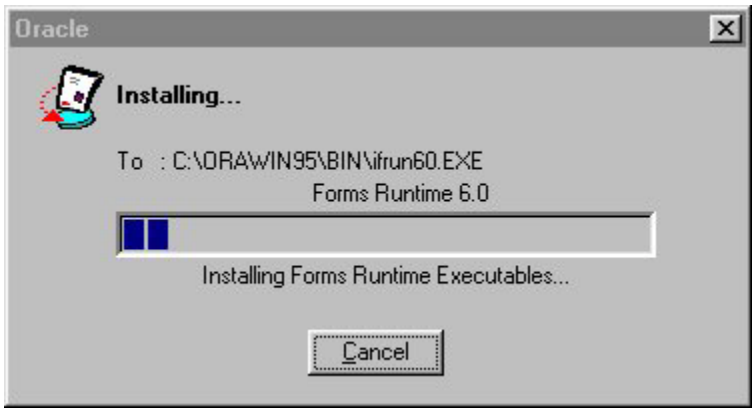


Figure 23 - Installation Progress

9. When you receive the message that the Installation was successful (Figure 24), click OK. The Software Asset Manager will again appear with the Developer components installed (Figure 25).
10. Click Exit to close the Software Asset Manager.

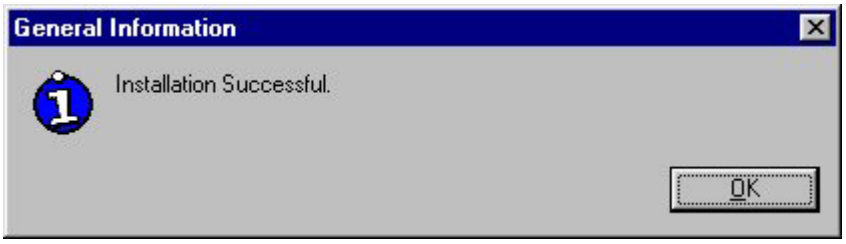


Figure 24 - Successful installation

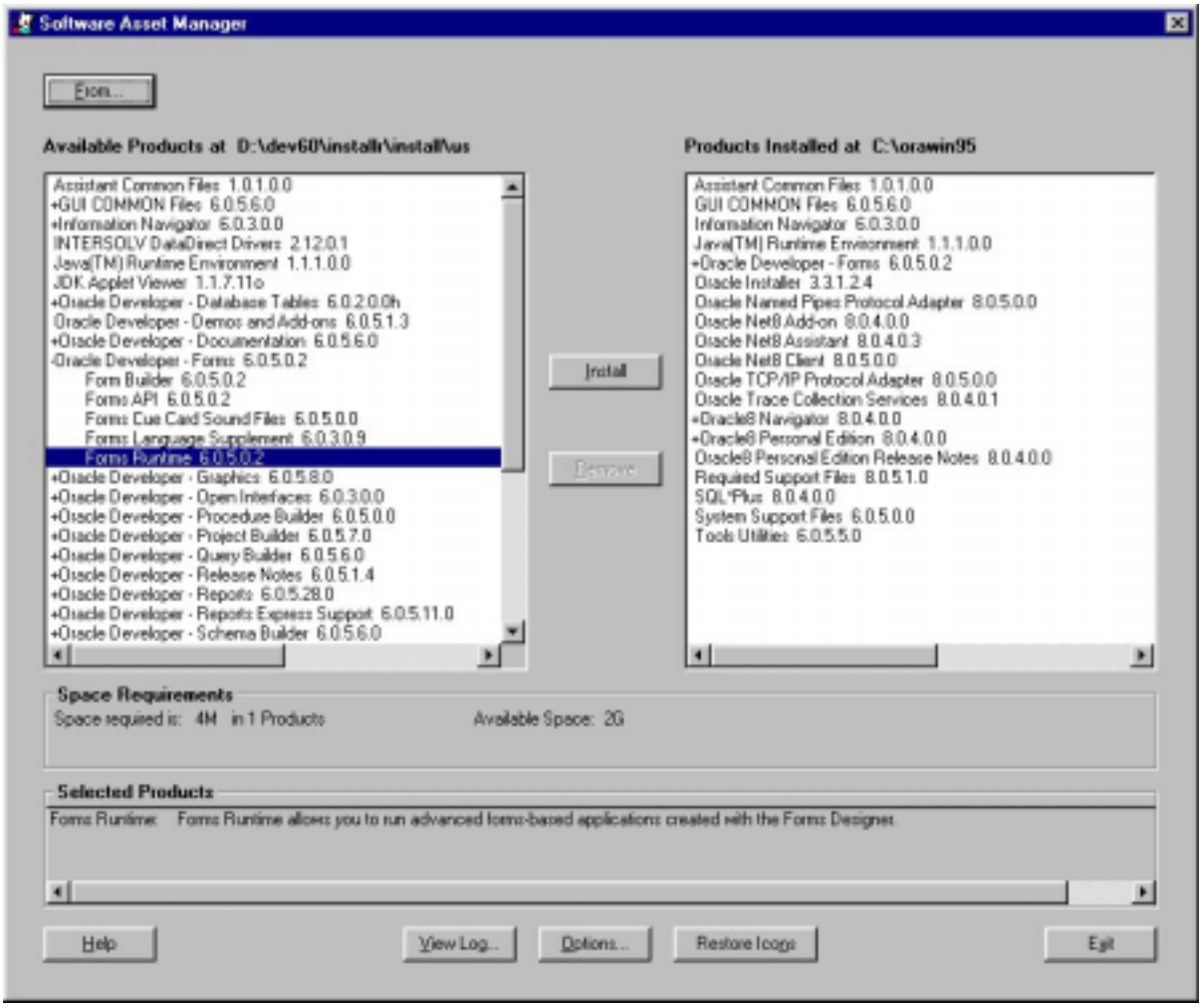


Figure 25 - Software Asset Manager after Developer Installation

Install SQR Workbench

1. Insert the SQR CD in the CD drive. If the autorun feature fails to start the installation, navigate to the CD and double-click Setup.exe (Figure 26) or run Setup from Run on the Windows Menu (Figure 27).
2. The installation will be in three parts, each part will be installed prior to the start of the next.
3. When asked for the Installation Key, enter the following:

   12117-0026-6139-4927-1540-7882-5194-5796-5026

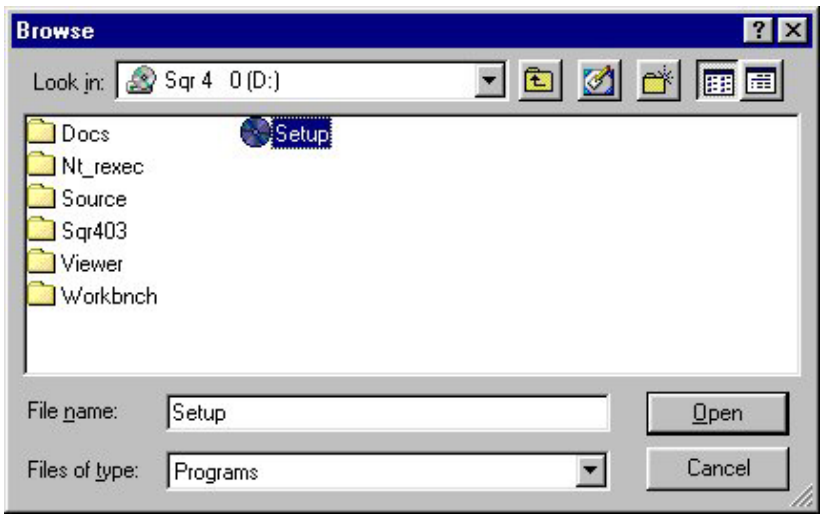


Figure 26 - CD path to Setup

Figure 27 - Run dialog

4. Select the products shown and click Install.

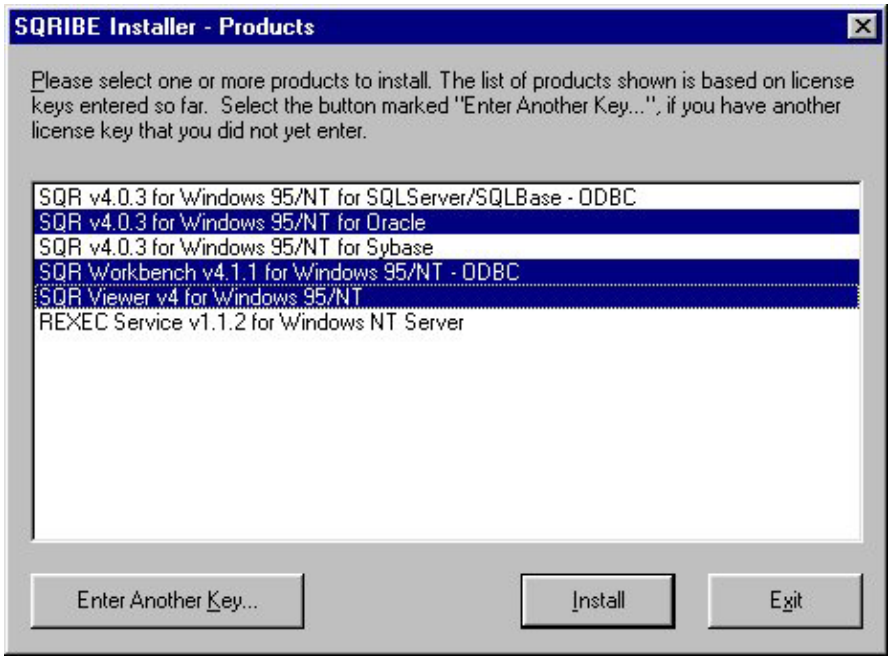


Figure 28 - SQRIBE Installer Products List

5. Enter the Serial Number as shown and click Next to Continue (Figure 29). When the setup is complete, click Finish (Figure 30) to continue with the next installation (Figure 31).

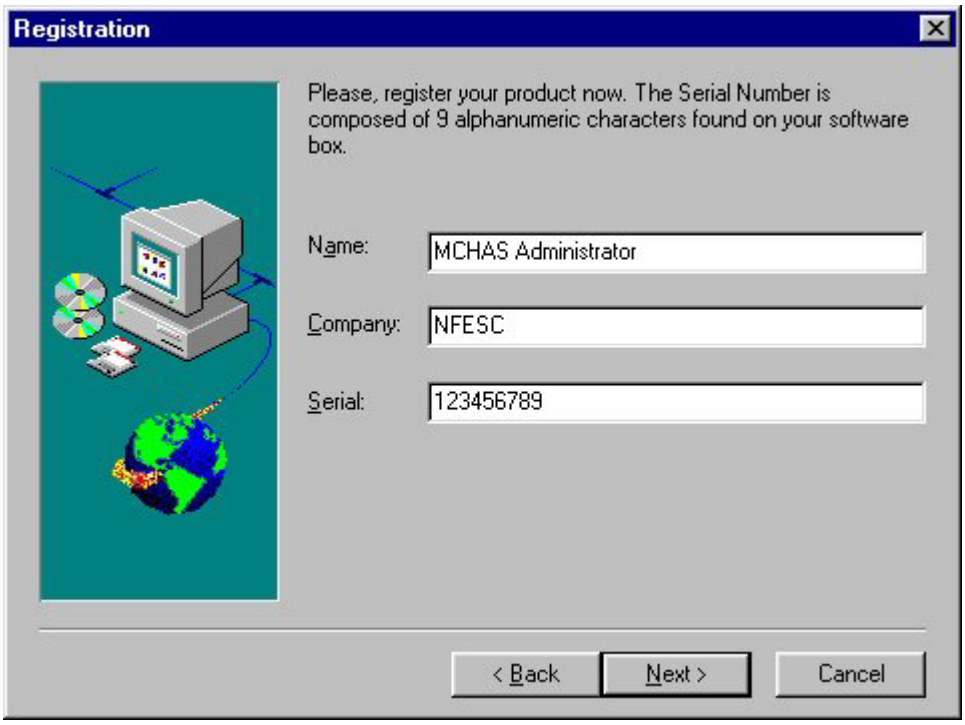


Figure 29 - Registration

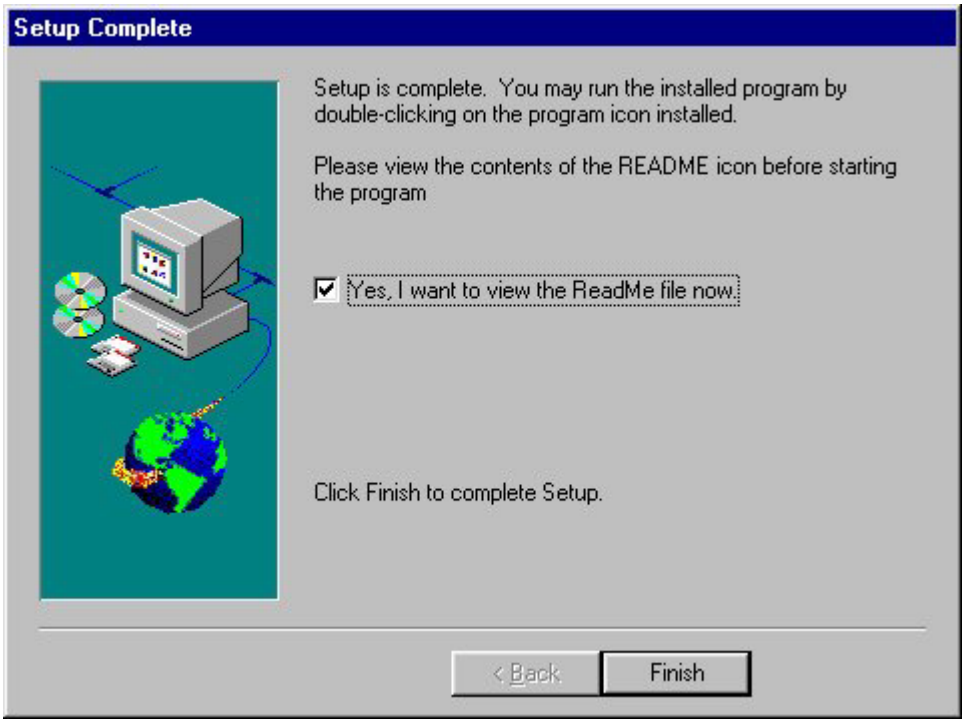


Figure 30 - Part I Setup Complete

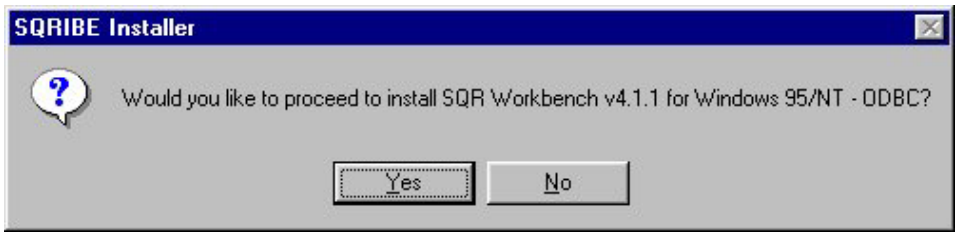


Figure 31 - Part II Installation

6. Select the Type of Installation (Figure 32) and the components (Figure 33) to be installed. When setup is complete (Figure 34), click Finish to begin setting up your ODBC driver (Figure 35).

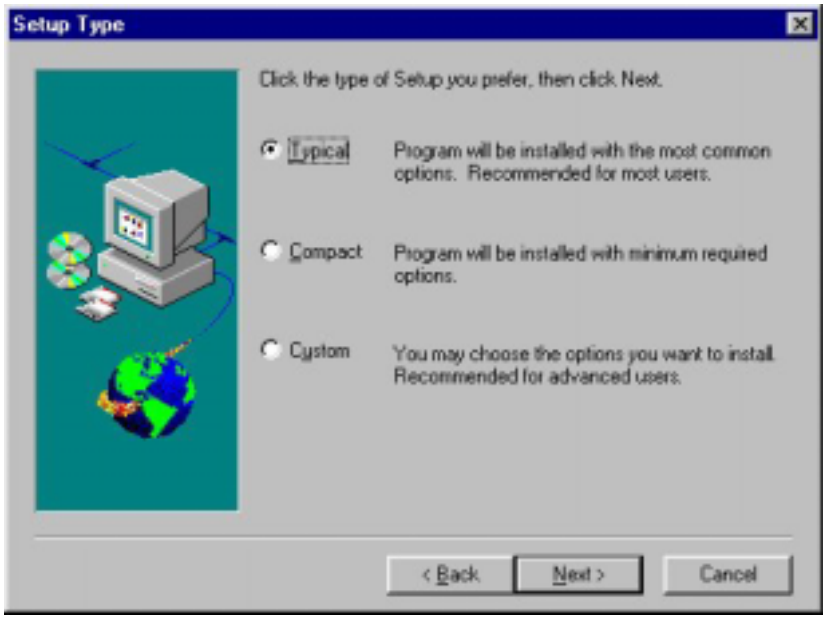


Figure 32 - Setup Type dialog

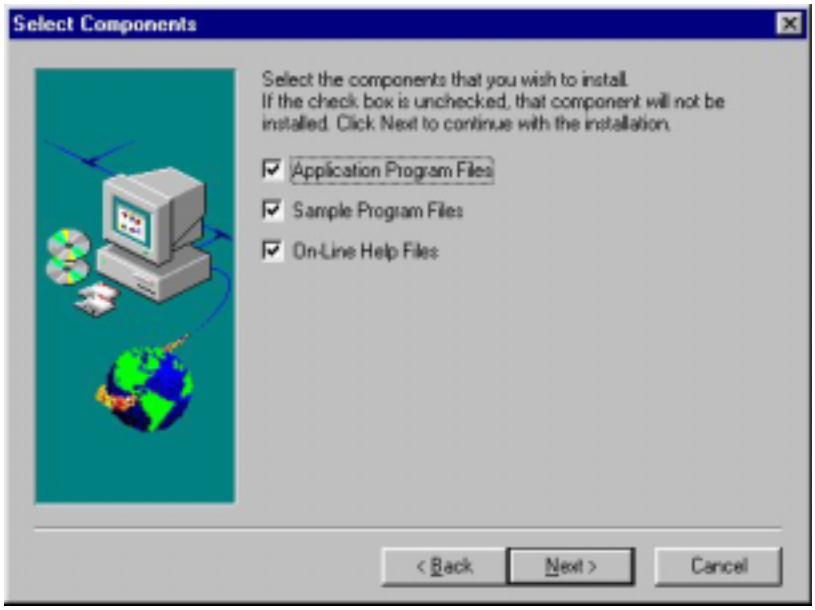


Figure 33 - Setup Components

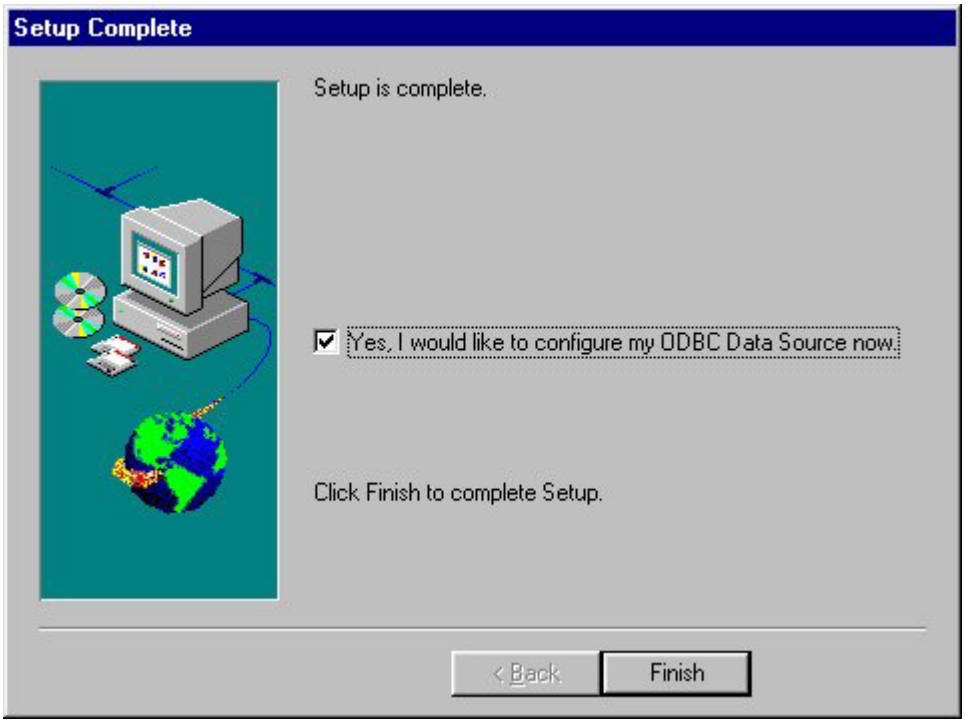


Figure 34 - Part II Setup Complete

Figure 35 - ODBC dialog

7. When the ODBC Data Source Administrator appears (Figure 36), click the System DSN tab.
8. Click Add and select the Oracle ODBC Driver from the Create a New Data Source window (Figure 37).
9. Complete the Setup window as shown (Figure 38).
10. The ODBC Data Source Administrator will now show your new connection (Figure 39) click OK to finish and begin Part III of the SQR Installation (Figure 40).

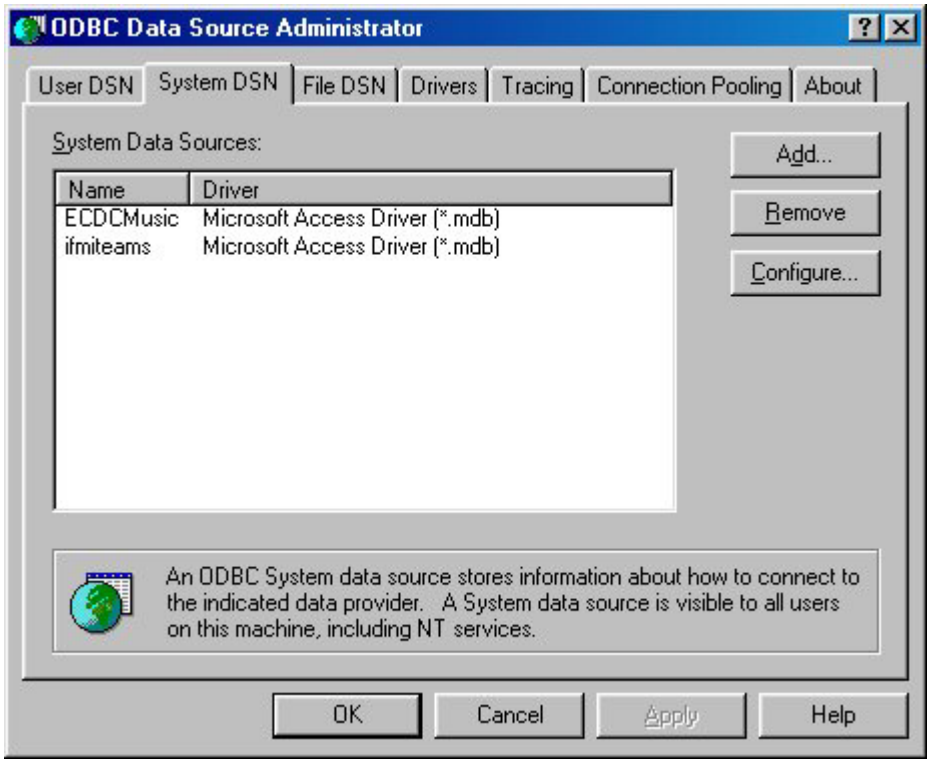


Figure 36 - ODBC Administrator

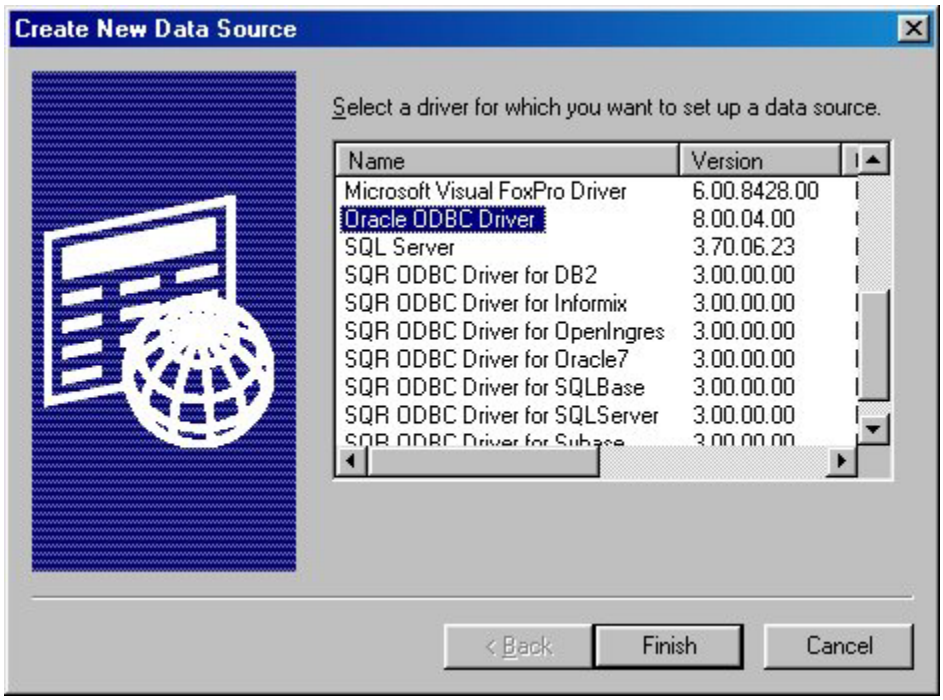


Figure 37 - Create New Data Source

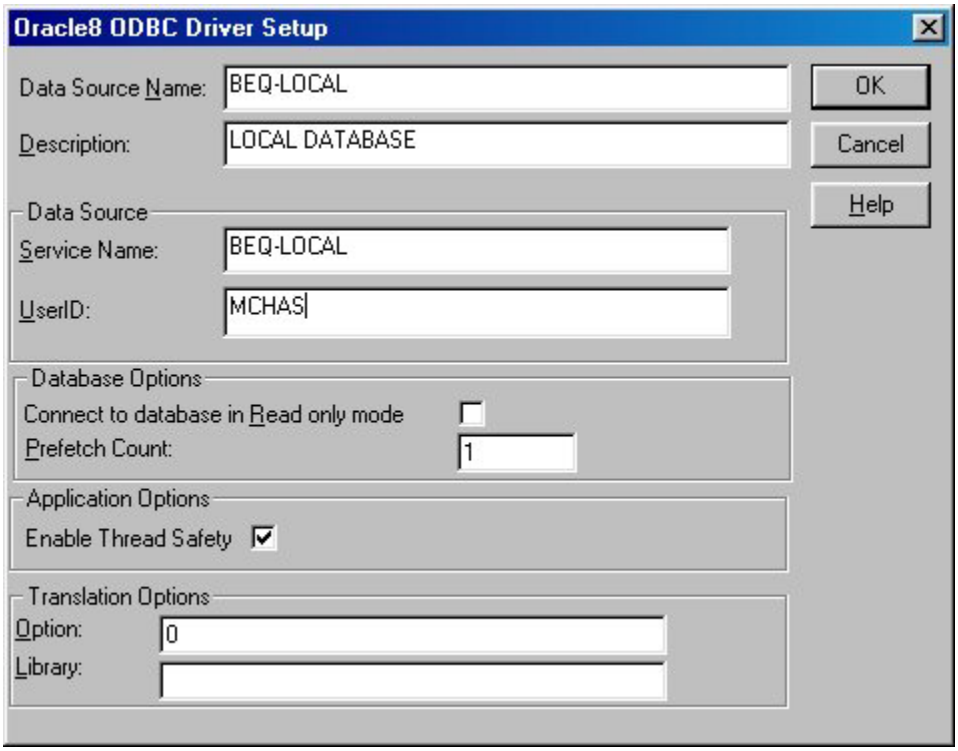


Figure 38 - Setup ODBC Driver

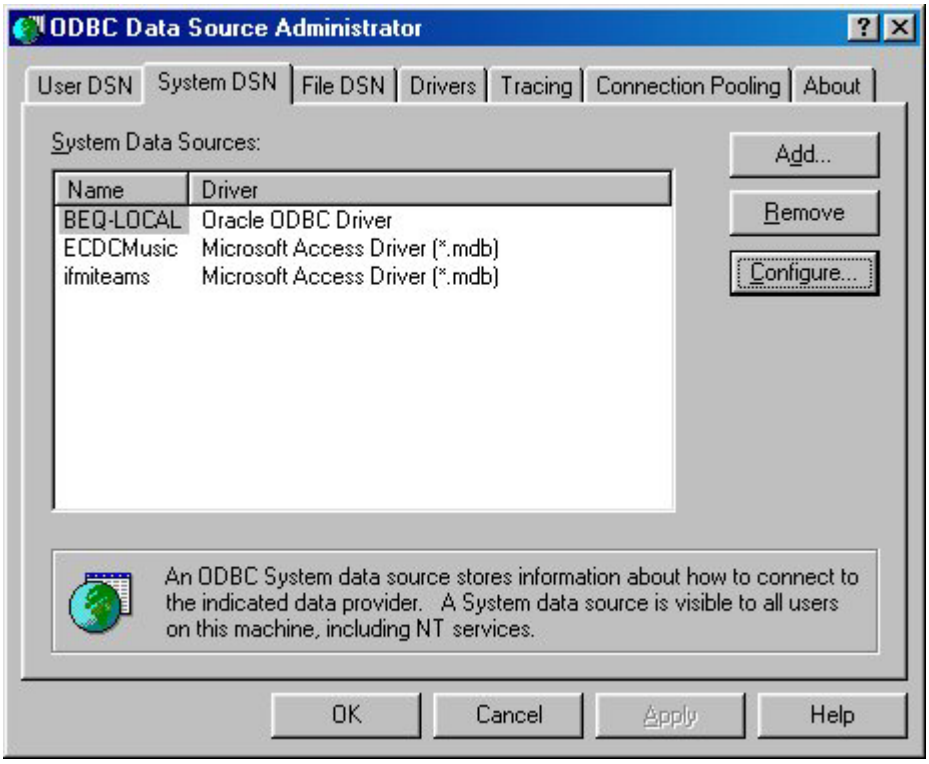


Figure 39 - ODBC Setup Complete

Figure 40 - Part III Installation

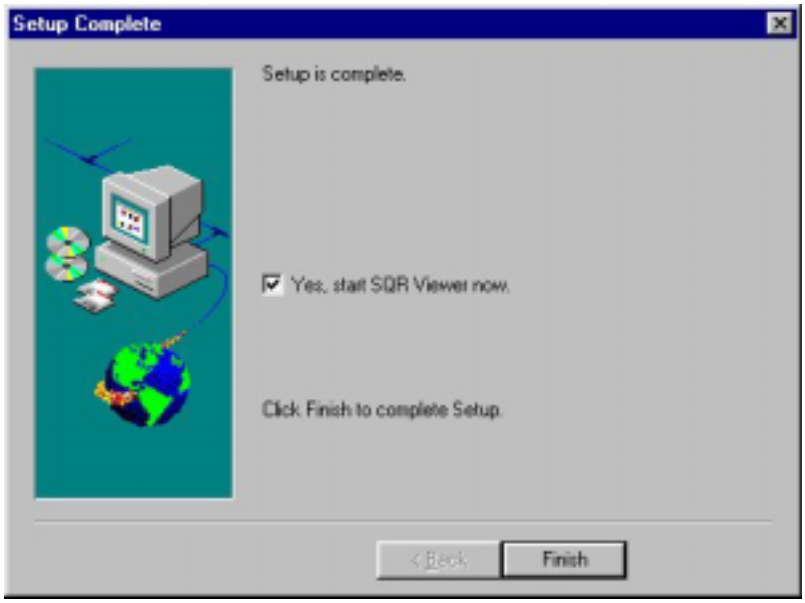


Figure 41 - Part III Setup Complete

11. When the Installation is complete (Figure 41) click Finish. You may dismiss the Products List (Figure 42) by clicking OK.

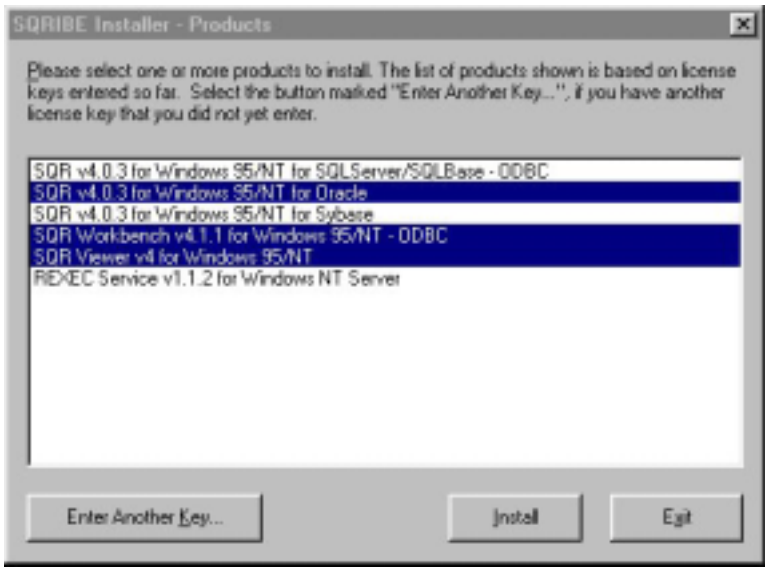


Figure 42 - Products List

Install MCHAS runtime forms

1. Copy all the executables to the local hard-drive.

Create the User MCHAS

1. Open SQL *Plus that was installed during the Oracle8 installation.
2. At the SQL Prompt (SQL>) type the following:

```
CREATE TABLESPACE MCHAS
DATAFILE 'c:\orawin95\database\mc25.ora' size 60m reuse
DEFAULT STORAGE (INITIAL 1000K NEXT 1000K PCTINCREASE 0 MAXEXTENTS 100);
CREATE TABLESPACE MCHAS_INDEX
DATAFILE 'c:\orawin95\database\mc25ind.ora' size 60m reuse
DEFAULT STORAGE (INITIAL 1000K NEXT 1000K PCTINCREASE 0 MAXEXTENTS 100);
```

3. Create the User MCHAS

```
CREATE USER MCHAS IDENTIFIED BY MCHAS
DEFAULT TABLESPACE MCHAS
TEMPORARY TABLESPACE TEMPORARY_DATA;
```

4. When you receive the message "User Created" type the following:

```
GRANT CONNECT, RESOURCE, DBA TO MCHAS;
```

Import the Database

1. This process will import an existing database export into your new database. This procedure will be done from the DOS prompt. From the Start menu, select Programs and navigate to and select MS-DOS Prompt.
2. When the DOS session window is open, type **IMP80 and ENTER** .

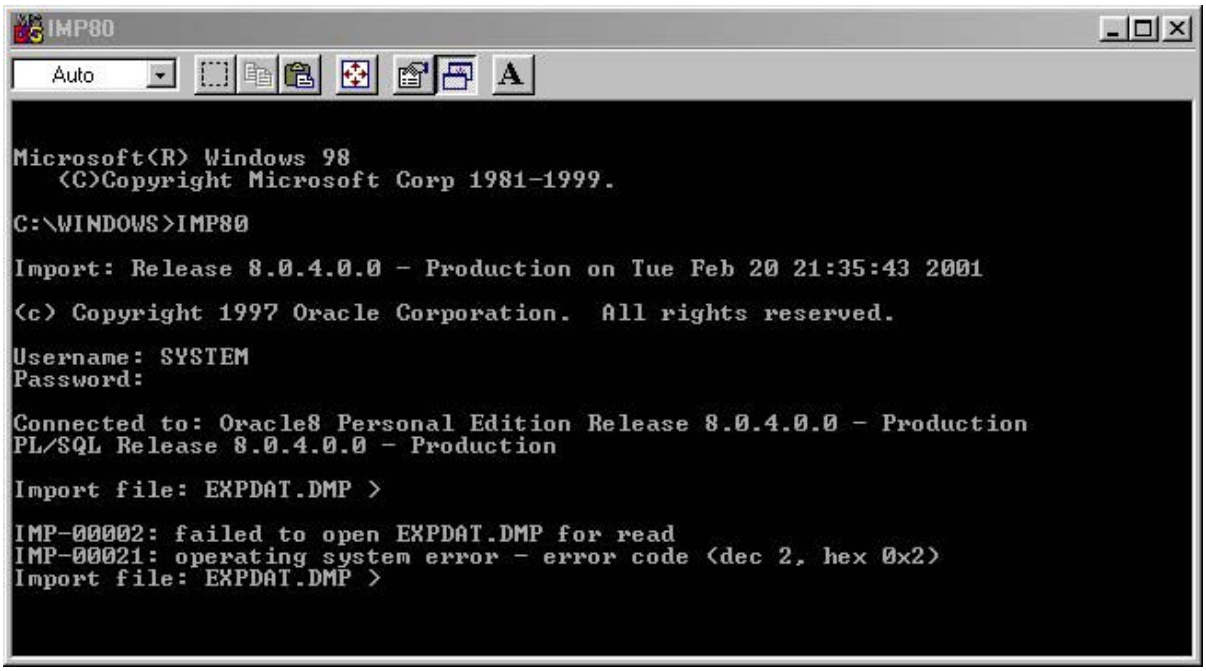


Figure 43 - MS-DOS Prompt

3. Enter the Username **SYSTEM** when prompted.
4. Enter the Password **MANAGER** when prompted. (Nothing will appear as you type the password).
5. Type the path to the exported database including file name.
6. Follow the prompts instructions.

Modify Configuration Files

1. Modify the Registry Setting (this step requires that you have administrative rights to the local PC running NT)
   A. Run REGEDIT
   B. Navigate to the following key:
   C. My Computer\HKEY_LOCAL_MACHINE\Software\Oracle
   D. Add or Modify the value as follows where 'X' = Your local drive

   FORMS60_PATH = X:\MCHAS\FORMS60;X:\MCHAS\PLSQLLIB

   REPORT60_PATH= X:\MCHAS\REPORT60;X:\MCHAS\PLSQLLIB

   UI_ICON=X:\MCHAS\ICONS

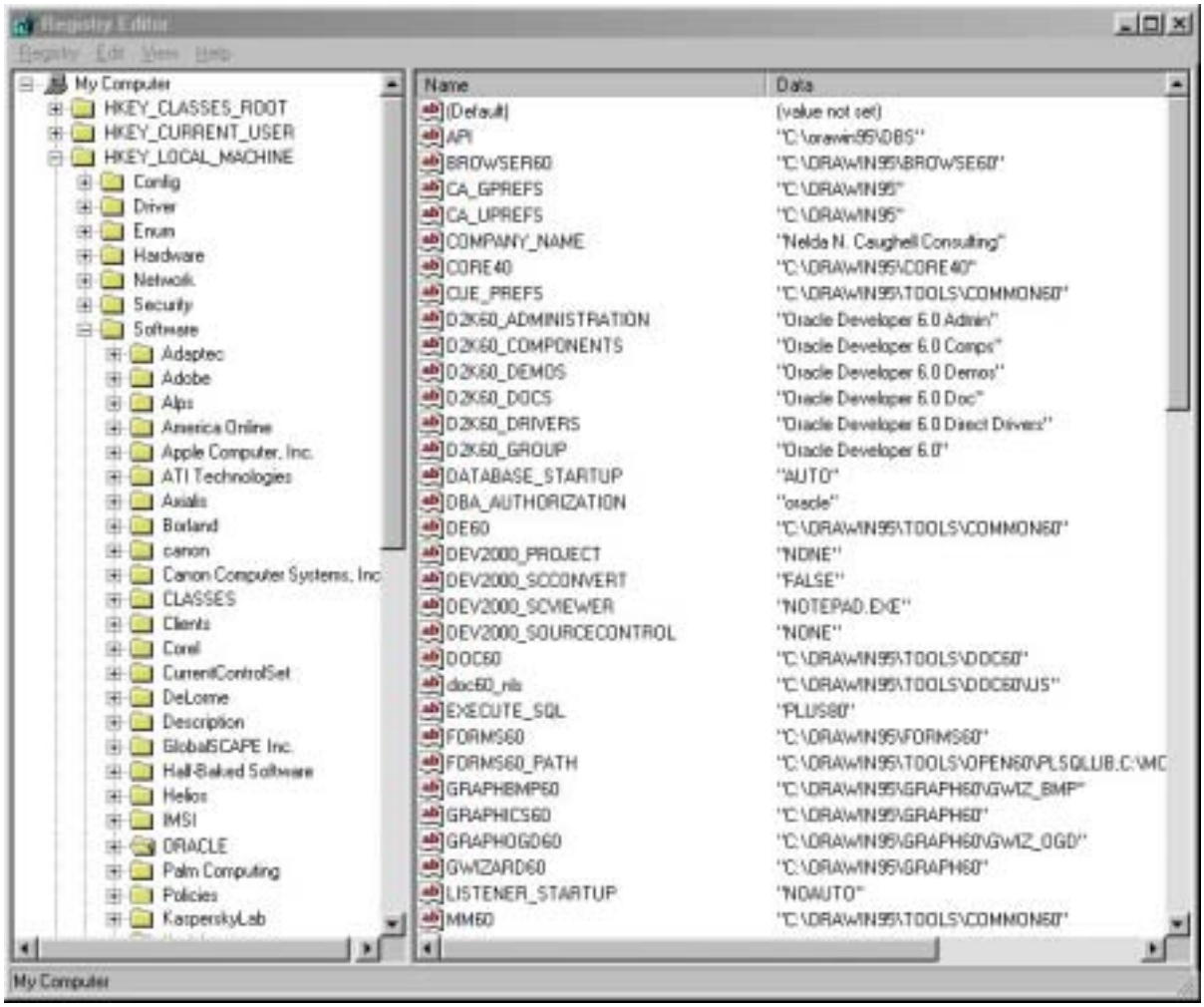


Figure 44 - Windows Registry

2. Modifying WIN.INI configuration file
   A. Add the following section if it does not currently exist, where 'X' = Your local drive:

      [SQR Products]

      SQRDIR=X:\SQR\ORA\BINW
3. Modifying SQR.INI configuration file
   A. Modify the following sections, where 'X' = Your local drive:

      [Environment:common]

      SQRDIR=X:\SQR\ODB\BINW

      SQRFLAGS=-KEEP -XCB -PRINTER:LP

      SQRW_SPOOL=X:\SPL

      and

      [Environment:Oracle]

      SQRDIR=X:\SQR\ODB\BINW

      SQRFLAGS=-KEEP -XCB -PRINTER:LP

      SQRW_SPOOL=X:\SPL

Creating a MCHAS backup

Creating a backup of your databse consists of exporting the tables and data. This is done through an MC-DOS window.

1. From the **Start Menu**, select **Programs**, select **MS-DOS Prompt**.
2. You may navigate to another directory using **DOS** commands. At the prompt, type EXP80.
3. Type in the Username and Password when prompted (Figure 45).

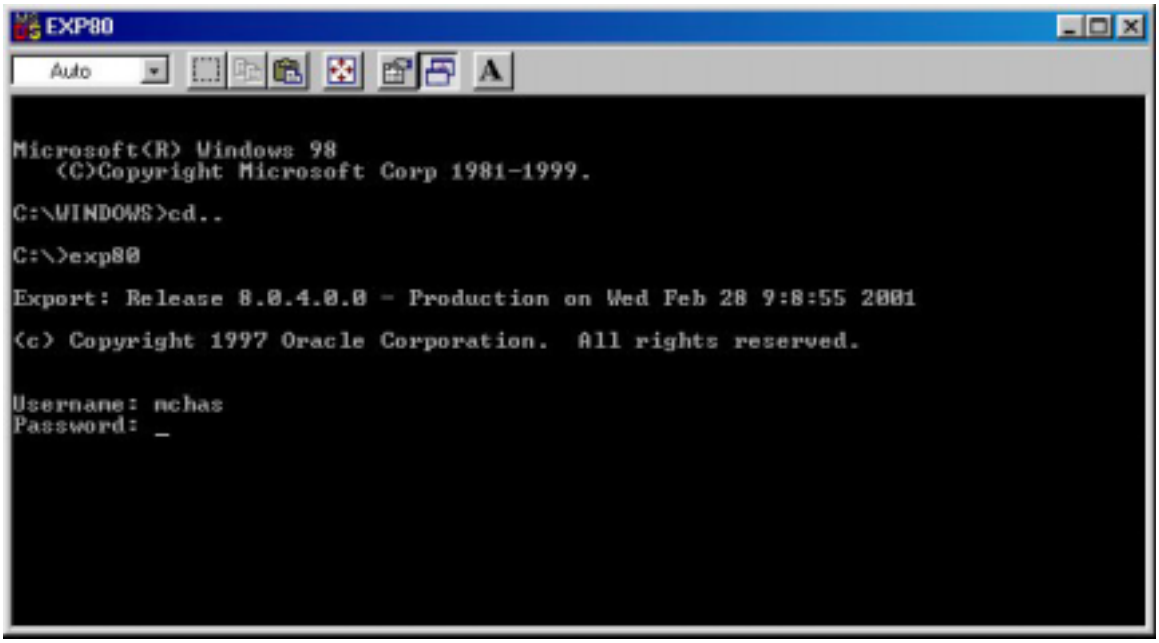


Figure 45 - Creating a Backup 1

4. Provide a name for your export file (Figure 46).

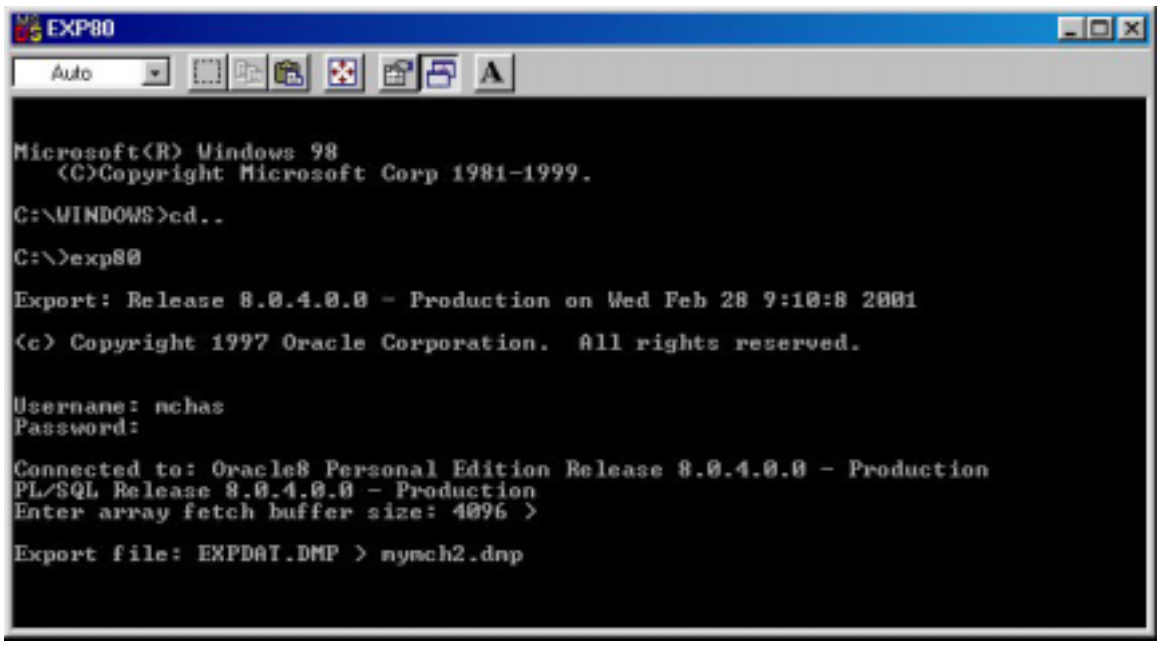


Figure 46 - Creating a Backup 2

You may export the entire database, only the users or only the tables. The default export option is shown following the **:** For purposes of this exercise, select the defaults, press enter at each line.

5. When notified "About to export specified users …" type the user name you wish to export, in this case, the user MCHAS, press enter. You will be prompted for another user, to export only the user MCHAS, press enter without entering a username (Figure 47).

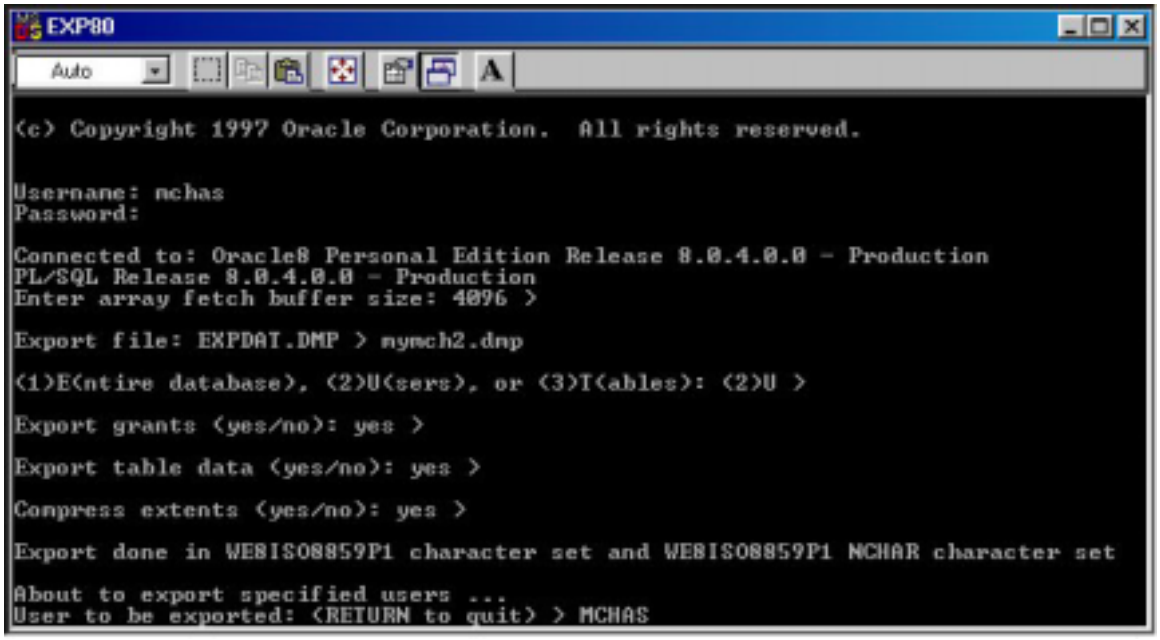


Figure 47 - Creating a Backup 3

Figure 48 - Creating a Backup 4

6. When the export is finished, type **EXIT** to close the MS-DOS window (Figure 48). If problems were encountered, you may open the orclALRT.log log-file created during the export to view the errors.

Recovering MCHAS from an Oracle Backup

Recovering from a backup imports the data and tables back from the export. See the Section "Import the Database" for step by step instructions.

Intro to SQL-Plus

SQL*Plus is the application used to communicate directly to the database. Open SQL *Plus by going to the Windows Start Menu > Programs > Oracle for Windows 95 and selecting SQL Plus 8.0.

Connecting to the Database

1. Enter the User Name, Password and Host String (Database Name) and press OK (Figure 50).

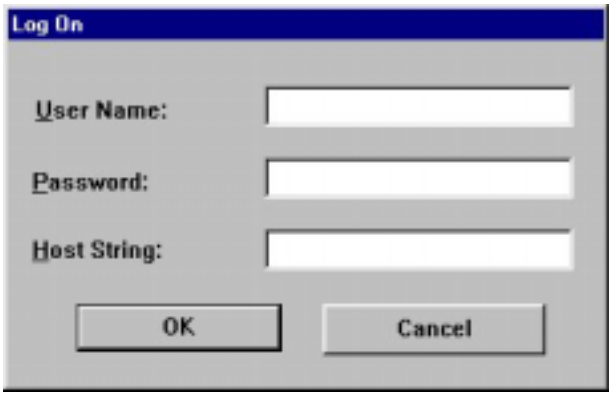


Figure 49 - SQL Plus log on

2. SQL *Plus will open with the prompt ready for input (Figure 50).

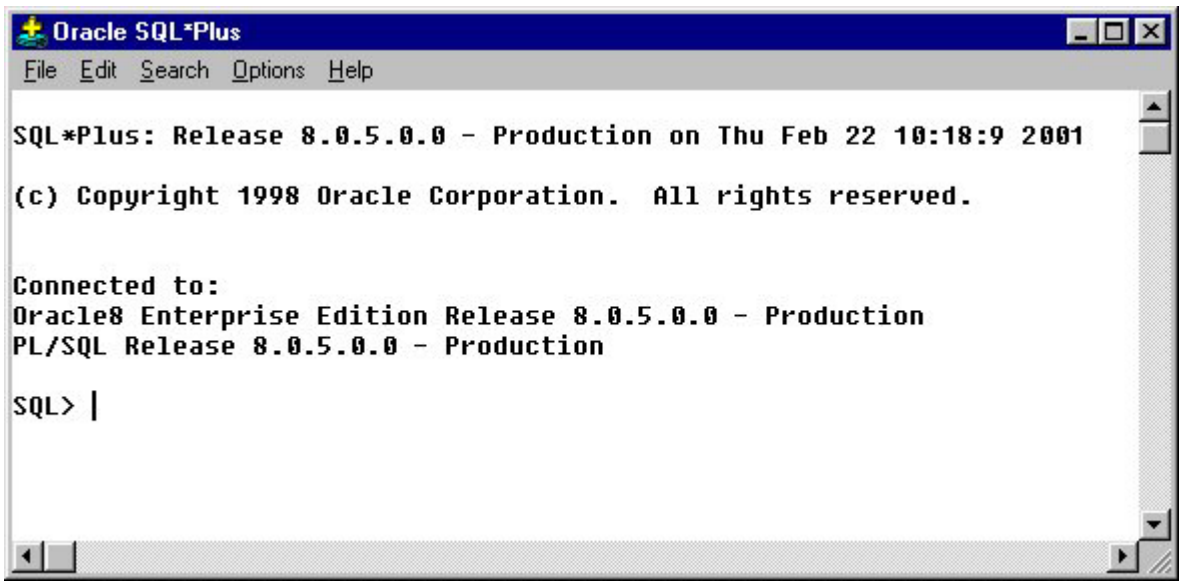


Figure 50 - SQL Plus

Setting the Environment

1. Select Options from the Oracle SQL *Plus Menu (Figure 50). Choose Environment and the Environment window will open (Figure 51).

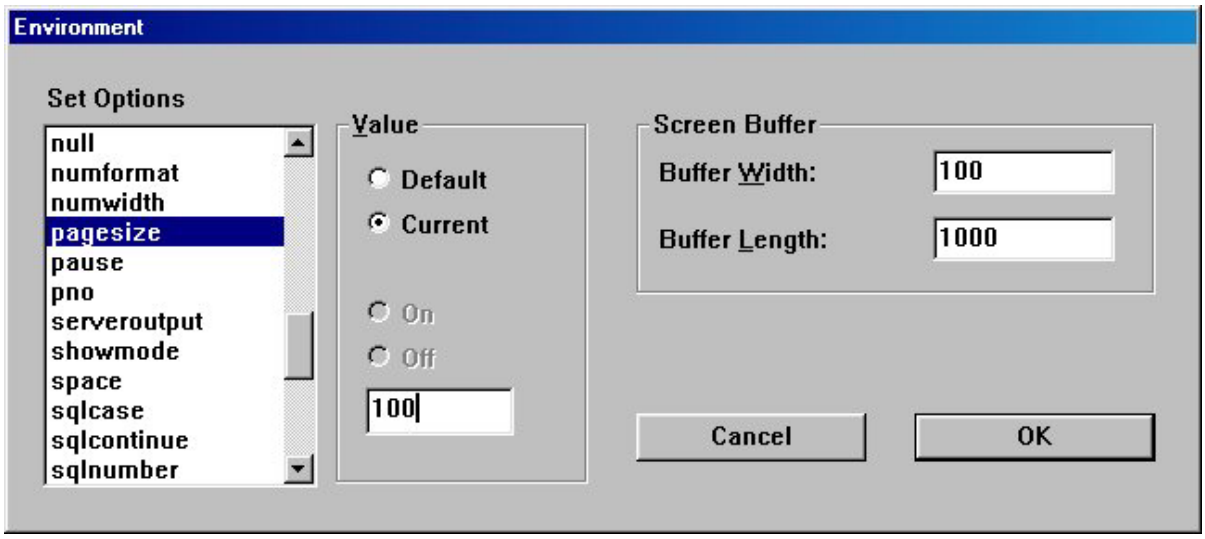


Figure 51 - SQL *Plus Environment Settings

2. Use this window to alter the SQL *Plus environment. (See the SQL *Plus documentation for a full description of these settings). Environment settings may also be set at the command line…`SQL>set pagesize 100.`

Spool Files

1. Spool files collect what is entered at the keyboard and returned from a fetch to the database and saves this information in an ASCII text file. To create a spool file, at the SQL> prompt, type:

```
SPOOL C:\SPL\MYSPOOL.SPL (The path will be where you want to save the
file).
```

2. When finished, type:

   ```
   SPOOL OFF;
   ```

3. The file can now be opened with any text editor, like Notepad.

Overview of an Oracle Database

SID

System Identifier

Table Spaces

A tablespace is an area of disk consisting of one or more disk files. A tablespace can contain many tables, indexes, or clusters. Because a tablespace has a fixed size it can get full as rows are added to its tables. The tablespace can be expanded by creating a new disk file and adding it to the tablespace or extending the existing datafiles.

Extents

Each table in the database has a single area of disk space called a *segment* set aside for it in the tablespace. Each segment, in turn, has an initial area of disk space set aside for it in the tablespace called the *initial extent*. Once it has used up this space, the *next extent*, another single area of disk space, is set aside for it. As this space is used, yet another extent is set aside.

Init(sid).ora file

Also known as the initialization parameter file. INIT*SID*.ORA is a database system file containing a list of parameters that are read when the database system (identified by the variable SID in the file name) is started. The parameters identify the database and control systemwide resources.

Rollback Segments

Allows transactions not committed to the database to be rolled back, or undone. Use with SAVEPOINT to control where the rollback will roll back to. SAVEPOINT is a point within a transaction to which you may rollback. Savepoints allow you to ROLLBACK portions of your current transaction.

Altering System Parameters

Initialization parameters can be grouped by function in several different ways. For example, there are parameters that perform the following functions:

- set database-wide limits
- set user or process limits
- name files or directories required by a database system
- set limits on database resources
- affect performance (these are called variable parameters)

The set of variable parameters are of particular interest to database administrators because these parameters are used primarily for improving database performance.

Intro to DML

Data Manipulation Language Statements.

Select Statement

A fetch from the database.

FORMAT:

```
SELECT [DISTINCT | ALL]
  { *
  | { [user.] table | view | snapshot} .*
    | expr [ [AS] column_alias] }
[, { [user.] table | view | snapshot} .*
  | expr [ [AS] column_alias] } ] ...
FROM { [user.] table [PARTITION (partition_name) | @dblink]
        |[view |snapshot] [@dblink]} [table_alias]
     | [THE] ( subquery ) [table_alias]
     | TABLE (nested_table_column) [table_alias]}
  [, { [user.] table [PARTITION (partition_name) | @dblink]
        |[view |snapshot] [@dblink]} [table_alias]
     | [THE] ( subquery ) [table_alias]
     | TABLE (nested_table_column) } ] ... [WHERE condition]
[ [ START WITH condition] CONNECT BY condition
  | GROUP BY expr [, expr] ... [HAVING condition] ] ...
  [ { UNION
    | UNION ALL
    | INTERSECT
    | MINUS} SELECT command]
[ ORDER BY expr | position | column_alias} [ASC | DESC]
      [, expr | position | column_alias} [ASC | DESC] ] ...
| FOR UPDATE [OF [ [user.] table. | view.} ] column
          [, [ [user.] table. | view.} ] column] ...] [NOWAIT]____
```

SAMPLE:

```
select * from occupant;
select last_name, first_name from occupant where occupant_ssn =
'123456789';
```

Insert Statement

Inserting data into a table in the database.

FORMAT:

```
INSERT INTO [USER.]TABLE[@DB_LINK] [(COLUMN [,COLUMN]…)]
  VALUES (EXPRESSION [.EXPRESSION]…) | QUERY…);
```

SAMPLE:

```
Insert into inspector values ('GM102','STOVALL','STANLEY','P','GM');
```

Update Statement

Update, or modify data in a table.

FORMAT:

```
UPDATE [user.]table[@dblink]
SET { column = expression | column = (select expression...)
```

```
    [ [,column = expression]... |
      [,column = (select
expression...)]... ] |
    (column [,column]...) = (subquery)}
[WHERE {condition | CURRENT OF cursor}];
```

SAMPLE:

```
Update inspector set proj_area_cd = 'GMO'
Where proj_area_cd = 'GM';
```

Delete Statement

Delete data from a table that satisfies a condition.

FORMAT:

```
DELETE FROM [user.]table[@link] [alias] [ WHERE condition ];
```

SAMPLE:

```
Delete from inspector where inspector_id = 'GM102';
```

Intro to DDL

Data definition Language Statement.

Database Objects

Users – Persons with access to the database.

Tables – Hold values in a matrix or spreadsheet-like format.

Views – Customized look at data from one or more tables.

Indexes – Prioritizes seeks to a table.

Sequences – The automatic progression of a number.

Packages and Procedures

Package: A package is a collection of related procedures, functions, and other program objects stored together in the database.

Procedure: A set of SQL and PL/SQL statements that are grouped together as a unit to solve a specific problem or perform a set of related tasks.

Synonyms – A name given to a database object to be commonly used by users to access the object.

Triggers – Performs an action when another action occurs.

Constraints – A limitation set on a table.

Creating Database Objects

User

FORMAT:

```
CREATE USER user IDENTIFIED {BY password | EXTERNALLY}
[DEFAULT TABLESPACE tablespace]
[TEMPORARY TABLESPACE tablespace]
```

```
[QUOTA integer {K | M} | UNLIMITED} ON tablespace]
[PROFILE profile]
| PASSWORD EXPIRE
| ACCOUNT { LOCK | UNLOCK}
```

SAMPLE:

```
create user STOVALLSP identified by PASSWORD
default TABLESPACE MCHAS
TEMPORARY TABLESPACE TEMPORARY_DATA;
```

Tables

FORMAT:

```
Relational table definition ::=
CREATE TABLE [user.] table
  [({ column datatype [DEFAULT expr] [WITH ROWID]
      [SCOPE IS [user.] scope_table_name] [column_constraint] ...
   | table_constraint | REF (ref_column_name) WITH ROWID
   | SCOPE FOR (ref_column_name) IS [user.]scope_table_name}
[, {column datatype [DEFAULT expr] [WITH ROWID]
      [SCOPE IS [user.] scope_table_name] [column_constraint] ...
   | table_constraint | REF (ref_column_name) WITH ROWID
   | SCOPE FOR (ref_column_name) IS [user.]scope_table_name} ] ...) ]
[ {[ORGANIZATION {HEAP | INDEX}
   | PCTTHRESHOLD [INCLUDING column_name]
     [OVERFLOW [physical_attributes_clause
        | TABLESPACE tablespace] ...]
   | physical_attributes_clause
   | TABLESPACE tablespace
     | LOB (lob_item [, lob_item ...]) STORE AS
     [lob_segname]
     [( TABLESPACE tablespace
        | STORAGE storage
        | CHUNK integer
        | PCTVERSION integer
        | CACHE
        | NOCACHE LOGGING | NOCACHE NOLOGGING
        | INDEX [lob_index_name]
           [( TABLESPACE tablespace
           | STORAGE storage
           | INITRANS integer
           | MAXTRANS integer ) ...])]
   | NESTED TABLE nested_item STORE AS storage_table
   | {LOGGING | NOLOGGING}] ...
| CLUSTER cluster (column [, column] ...)}]
[PARALLEL parallel_clause]
[PARTITION BY RANGE (column_list)
   (PARTITION [partition_name] VALUES LESS THAN (value_list)
   [physical_attributes_clause
   | TABLESPACE tablespace
   | {LOGGING | NOLOGGING}] ...
[ ENABLE enable_clause | DISABLE disable_clause] ...
   [AS subquery]
   [CACHE | NOCACHE]
physical_attributes_clause ::=
   [ PCTFREE integer
   | PCTUSED integer
   | INITRANS integer
   | MAXTRANS integer
```

```
   | STORAGE storage_ ]

Object table definition ::=
CREATE TABLE [user.]table OF [user.]object_type
   [( [column | attribute [DEFAULT expr] [WITH ROWID]
      [SCOPE IS [user.] scope_table_name][column_constraint] ...]
   | table_constraint | REF (ref_column_name) WITH ROWID
   | SCOPE FOR (ref_column_name) IS [user.]scope_table_name}
[, {column | attribute [DEFAULT expr] [WITH ROWID]
      [SCOPE IS [user.] scope_table_name] [column_constraint] ...
   | table_constraint | REF (ref_column_name) WITH ROWID
   | SCOPE FOR (ref_column_name) IS [user.]scope_table_name} ] ...) ]
   [OIDINDEX [index] [( physical_attributes_clause
             | TABLESPACE tablespace) ...]
[ { [physical_attributes_clause
   | TABLESPACE tablespace
   | LOB (lob_item [, lob_item ...]) STORE AS
     [lob_segname]
     [( TABLESPACE tablespace
        | STORAGE storage
        | CHUNK integer
        | PCTVERSION integer
        | CACHE
        | NOCACHE LOGGING | NOCACHE NOLOGGING
        | INDEX [lob_index_name]
          [( TABLESPACE tablespace
          | STORAGE storage
          | INITRANS integer
          | MAXTRANS integer )] )]
  | NESTED TABLE nested_item STORE AS storage_table
  | {LOGGING | NOLOGGING} ] ...
| CLUSTER cluster (column [, column] ...)}]
[PARALLEL parallel_clause]
[ ENABLE enable_clause
| DISABLE disable_clause] ...
[AS subquery]
[CACHE | NOCACHE]
```

SAMPLE:

```
CREATE TABLE APPLICATIONS (
NAME                VARCHAR2(10) NOT NULL,
DESCRIPTION         VARCHAR2(30) NOT NULL,
DEFAULT_PASSWORD    VARCHAR2(20) NOT NULL,
DEFAULT_TABLESPACE  VARCHAR2(30) NOT NULL,
TEMP_TABLESPACE     VARCHAR2(30) NOT NULL )
PCTUSED 40 PCTFREE 10 TABLESPACE MCHAS STORAGE ( INITIAL 10240 NEXT 10240
PCTINCREASE 0 MAXEXTENTS 100);
```

Views

FORMAT:

```
CREATE VIEW [OR REPLACE] [FORCE | NO FORCE] VIEW [user.] view
   [ [(alias [alias] ...)]
    | OF [user.]type_name WITH OBJECT OID [DEFAULT
                        | (attribute [, attribute] ...)] ]
   AS subquery
   [WITH [ READ ONLY
           | CHECK OPTION [CONSTRAINT constraint] ] ]
```

SAMPLE:

```
create or replace view CLASS ( occupant_ssn ) as select distinct
occupant_ssn from unit_activity;
```

Indexes

FORMAT:

```
CREATE [UNIQUE | BITMAP] INDEX [user.] index
  ON { [user.] table (column [ASC | DESC]
             [, column [ASC | DESC] ] ...)
  | CLUSTER [user.] cluster}
  [ physical_attributes_clause
  | {LOGGING | NOLOGGING}
  | {TABLESPACE tablespace | DEFAULT}
  | {NOSORT | REVERSE} ] ...
  [ GLOBAL PARTITION BY RANGE (column_list)
     (PARTITION [partition_name]
       VALUES LESS THAN (value_list)
     [ physical_attributes_clause
     | {LOGGING | NOLOGGING} ], ...)
  | LOCAL [(PARTITION [partition_name]
     [ physical_attributes_clause
     | {LOGGING | NOLOGGING} ], ...) ] ]
[ PARALLEL parallel_clause ]
physical_attributes_clause ::=
   [ PCTFREE integer
   | PCTUSED integer
   | INITRANS integer
   | MAXTRANS integer
   | STORAGE storage_]
```

SAMPLE:

```
create index wl_idx on wait_list (waitlist_cd)
tablespace mchas_index;
create index wl_pos on wait_list (pos)
tablespace mchas_index;
```

Synonyms

FORMAT:

```
CREATE [PUBLIC] SYNONYM [user.]synonym
   FOR [user.]table [@database_link];
```

SAMPLE:

```
create public synonym HOUSING_UNIT         for HOUSING_UNIT;
create public synonym HOUSING_UNIT_GRADE   for HOUSING_UNIT_GRADE;
create public synonym INSPECTOR            for INSPECTOR;
create public synonym INSPECTION           for INSPECTION;
create public synonym OCCUPANT             for OCCUPANT;
```

Triggers

FORMAT:

```
CREATE [OR REPLACE] TRIGGER [user.]trigger
   {BEFORE | AFTER | INSTEAD OF}
   {DELETE | INSERT | UPDATE [OF column[, column]...}
   [OR {DELETE | INSERT | UPDATE [OF column[, column]...}]...
   ON [user.]table
   [REFERENCING {OLD [AS] old | NEW [AS] new}]
   [FOR EACH ROW]
   [WHEN (condition)]
   block
```

SAMPLE:
```
create or replace trigger ins_occupant_t
before insert on occupant
for each row
begin
:new.insert_by := user;
:new.insert_dt := sysdate;
:new.app_dt := sysdate;
end;
.
/
create or replace trigger upd_occupant_t
before update on occupant
for each row
begin
:new.update_by := user;
:new.update_dt := sysdate;
end;
.
/
```

Altering Database Objects

Users

You use ALTER USER to change a user's password, or the DEFAULT tablespace or TEMPORARY tablespace.

FORMAT:
```
ALTER USER user
  { IDENTIFIED {BY password | EXTERNALLY
           | GLOBALLY AS 'CN=user'}
  | DEFAULT TABLESPACE tablespace
  | TEMPORARY TABLESPACE tablespace
  | QUOTA { integer [K | M]
           | UNLIMITED} ON tablespace
   [QUOTA { integer [K | M]
           | UNLIMITED} ON tablespace] ...
| PROFILE profile
| PASSWORD EXPIRE
| ACCOUNT { LOCK | UNLOCK }
| DEFAULT ROLE { role [, role] ...
           | ALL [EXCEPT role [, role] ...]
            | NONE} } ..
```

SAMPLE:
```
ALTER USER STOVALLSP IDENTIFIED BY NEWPASSWORD
```

Tables

FORMAT:
```
ALTER TABLE [user.] table
   {ADD ( [ {column datatype [DEFAULT expr] [WITH ROWID]
           [SCOPE IS [user.] scope_table_name] [column_constraint] ...
         | table_constraint | REF (ref_column_name) WITH ROWID
         | SCOPE FOR (ref_column_name) IS [user.]scope_table_name ...) } ]
 | MODIFY ( [column [datatype] [DEFAULT expr]
           [column_constraint] ...] ...)
  | [physical_attributes_clause]
```

```
  | {LOB_storage_clause [, LOB_storage_clause...]}
  | {MODIFY_LOB_storage_clause [, MODIFY_LOB_storage_clause]}
  | {NESTED_TABLE_storage_clause [, NESTED_TABLE_storage_clause...]}
  | DROP drop_clause
  | ALLOCATE EXTENT [ ( {SIZE integer [K | M]
             | DATAFILE 'filename'
             | INSTANCE integer} ...) ]
  | DEALLOCATE UNUSED [KEEP integer [K | M] ] } ...
  [ { [ENABLE {enable_clause | TABLE LOCK}
     | DISABLE {disable_clause | TABLE LOCK} ] ...
   | [PARALLEL parallel_clause] {NOCACHE | CACHE} ]
 | RENAME TO new_table_name
 | OVERFLOW {physical_attributes_clause | INCLUDING column_name
        | ALLOCATE EXTENT [ ( {SIZE integer [K | M]
                     | DATAFILE filename
                     | INSTANCE integer} ...) ]
        | DEALLOCATE UNUSED [KEEP integer [K | M] ] } ...
 | ADD OVERFLOW [ {physical_attributes_clause
              | PCTTHRESHOLD integer
              | INCLUDING column_name
              | TABLESPACE tablespace} ...]
 | MODIFY PARTITION partition_name { physical_attributes_clause
                      | [LOGGING | NOLOGGING] } ...
 | MOVE PARTITION partition_name { physical_attributes_clause
                      | [LOGGING | NOLOGGING]
                      | TABLESPACE tablespace
                      | PARALLEL parallel_clause} ...
 | ADD PARTITION [new_partition_name]
     VALUES LESS THAN (value_list) { physical_attributes_clause
                      | [LOGGING | NOLOGGING]
                      | TABLESPACE tablespace } ...
 | DROP PARTITION partition_name
 | TRUNCATE PARTITION partition_name [DROP STORAGE | REUSE STORAGE]
 | SPLIT PARTITION partition_name_old AT (value_list)
   [INTO ( PARTITION [split_partition_1] [physical_attributes_clause
                      | [LOGGING | NOLOGGING]
                      | TABLESPACE tablespace ] ...
          , PARTITION [split_partition 2] [physical_attributes_clause
                      | [LOGGING | NOLOGGING]
                      | TABLESPACE tablespace ] ...) ]
              [ PARALLEL parallel_clause ] ...
 | EXCHANGE PARTITION partition_name
    WITH TABLE non_partitioned_table_name
     [{INCLUDING | EXCLUDING} INDEXES]
     [{WITH | WITHOUT} VALIDATION ]
 | MODIFY PARTITION UNUSABLE LOCAL INDEXES
 | MODIFY PARTITION REBUILD UNUSABLE LOCAL INDEXES }
physical_attributes_clause ::=
 [ PCTFREE integer
 | PCTUSED integer
 | INITRANS integer
 | MAXTRANS integer
 | STORAGE storage]
 LOB_storage_clause ::=
LOB (lob_item [, lob_item ...]) STORE AS
 [lob_segname]
   [( TABLESPACE tablespace
```

```
   | STORAGE storage_
   | CHUNK integer
   | PCTVERSION integer
   | CACHE
   | NOCACHE LOGGING | NOCACHE NOLOGGING
   | INDEX [lob_index_name]
      [( TABLESPACE tablespace
      | STORAGE storage
      | INITRANS integer
      | MAXTRANS integer ) ] )]
MODIFY_LOB_storage_clause ::=
  MODIFY LOB (lob_item)
     ( STORAGE storage
     | PCTVERSION integer
     | CACHE
     | NOCACHE LOGGING | NOCACHE NOLOGGING
     | ALLOCATE EXTENT [ ( {SIZE integer [K | M]
              | DATAFILE 'filename'
              | INSTANCE integer} ) ]
     | DEALLOCATE UNUSED [KEEP integer [K | M] ]
     | INDEX [lob_index_name]
     [ ( STORAGE storage
        | INITRANS integer
        | MAXTRANS integer
        | ALLOCATE EXTENT [ ( {SIZE integer [K | M]
                | DATAFILE 'filename'
                | INSTANCE integer} ) ]
        | DEALLOCATE UNUSED [KEEP integer [K | M] ] ) ]
NESTED_TABLE_storage_clause ::=
   NESTED TABLE nested_item STORE AS storage_table
```

SAMPLE:

```
alter table housing_unit add (washer_dryer      varchar2(1));
alter table housing_unit modify (classification varchar2(8));
```

Notes

Notes

Notes

System Administration

Modules and Components Overview

This information is being provided to System Administrators to aid in the setup of users for MCHAS 2.5a by giving insight into the Module/Component structure of the program.

Modules identify parts of the program to which not only access is given through components but also reports can be assigned to a Module. Access to individual pieces of the Module are also granted through components.

Modules for MCHAS 2.5a

MODULE NAME	MODULE DESCRIPTION
BADBOYS	Incident Reporting
CHARGE	Charges
EDITOCCU	Military Family
HSEUNTUT	Base Housing
INSPECT	Inspection Scheduling reports
LABELS	Printing of mailing labels
QRY_HSU	Custom Query reports for housing
QRY_REF	Generate Listings based on Query
RECEIPT	Receipts
REF_COPL	Referral Complexes
REF_UNIT	Off Base Referrals
SETASIDE	Set Aside Referrals
SYSADMIN	System Administration
TLA	TLA Tracking and Reports
UNACCOMP	Unaccompanied status and history
UTILIZAT	Utilization Reports
WAITLIST	Waitlists
WEAPONS	Weapons on base
18 Modules	

Components for MCHAS 2.5a

COMPONENT NAME	COMPONENT DESCRIPTION
APP_DT	Change an application date
A_HIST	Assignment History
BAHRATES	Modify the BAH Rate table
CHANGE_SSN	Change/Correct a sponsors SSN
CHARGE	Financial charges
CHARGE_POST	Posting of Financial Module recurring charges
CRLOG	Charge/Receipt log
DATA_CLEANUP	Execution of Data Cleanup
DSEARCH	Access to the Dependent Search screen
D_CHARGES	Setup the default charges table
EDITOCCU	Military Family Screen
EDIT_WAITLIST	Edit a person's waitlist info
FREEZE_ZONE	Freeze zone button
HSEUNTUT	Add and edit housing units
HUG	Edit the Housing-Unit-Grade table
INCIDENTS	On base incidents
INSPECTION	Inspection scheduling screen
INSPECTOR	Inspector screen
INST_INFO	Table of Marine base details
METER_ENTRY	Add and edit meters for houses
MET_INFO	See meter-reading data
MIL_SPOUSE	Swap Mil Spouse with sponsor
M_READ	Enter meter reading transactions
OWNER	Referral Owners
PAY_GRAD	Modify the pay-grade table
PICS	See and change pictures
PROJECT_CD	Edit the list of Project Area Codes
QRY_HSU	Housing Ad-hoc queries and reports
QRY_REF	Referral Ad-hoc queries and reports
RCPT_APP	Receipt Approval
RECEIPT	Receipt tracking
REF_COMPLEX	Add and edit complex listings

COMPONENT NAME	COMPONENT DESCRIPTION
REF_UNIT	Add and edit referral units
SEC_SETUP	Setup of user access to Mchas
SETASIDE	Set aside tracking and assign screens
SQR_SETUP	Add and edit SQR reports
SYSADMIN	Mchas administration screen
TIMELINE	Inspection scheduling timeline
TLA	Temporary Lodging Allowance
UA	Unit Activity Editing
UNACCOMP	Unaccompanied screen
UTIL_INPUT	Utilization Data Inputing
VALUELISTS	Maintain values in the valuelist table
WAITLIST	View wait lists
WAITLIST_CD	Edit the Waitlist_cd table
WAITLIST_DES	Edit the Waitlist_description table
WAIT_SEQ	Waitlist sequence button
WEAPONS	Maintain registered weapons for a sponsor
ZIP_CDS	Modify the zip-codes table
49 Components	

Quick Setup assignment for Components:

For use during the setup of Users, the following table outlines what components are granted during the "Quick Setup". Any Component can be granted to a User on an individual level. NOTE: If a component from System Admin is to be granted to a User, they must also be granted the Component "SYSADMIN" to allow them access to the screen through which the Admin component is accessed.

As you will see, Selecting ALL checkboxes will still only assign 45 of the 49 Components. These remaining four components are specialized functions that are assigned outside the normal for a Module.

Quick Setup	Components Granted
standard = 'Y'	Standard Components
	A_HIST
	DSEARCH
	EDITOCCU
	HSEUNTUT
	PICS
	TLA

Quick Setup	Components Granted
	UNACCOMP
	WAITLIST
housing = 'Y'	Housing Quarters Components
	HSEUNTUT
	QRY_HSU
	PROJECT_CD
referrals = 'Y'	Referral Units Components
	OWNER
	QRY_REF
	REF_COMPLEX
	REF_UNIT
	SETASIDE
waitlist = 'Y'	Waitlists Components
	EDIT_WAITLIST
	FREEZE_ZONE
	HUG
	WAIT_SEQ
	WAITLIST
	WAITLIST_CD
	WAITLIST_DES
financial = 'Y'	Financial Components
	BAHRATES
	CHARGE
	D_CHARGES
	M_READ
	MET_INFO
	METER_ENTRY
	RECEIPT
weapons = 'Y'	Weapons Tracking Components
	WEAPONS
incidents = 'Y'	Incident Reporting Components
	INCIDENTS
inspections = 'Y'	Inspection Scheduling Components
	INSPECTION

Quick Setup	Components Granted
	INSPECTOR
	TIMELINE
sysadmin = 'Y'	System Administration Components
	APP_DT
	CHANGE_SSN
	CRLOG
	INST_INFO
	INTEGRITY
	MIL_SPOUSE
	PAY_GRAD
	PROJECT_CD
	SEC_SETUP
	SQR_SETUP
	SYSADMIN
	UA
	VALUELISTS
	ZIP_CDS
Other Components	MUST be Assigned Individually Component Description (Module for Access)
CHARGE_POST	Post monthly recurring charges (Financial)
DATA_CLEANUP	Cleanup database (System Admin)
RCPT_APP	Approve Receipts (Financial)
UTIL_INPUT	Input Data – Exclusive Assignment Rights to/from Navy (Utilization)

Installation Info

MCHAS stores information about the military installation where it is installed. This information is to be used in utilization reports, to establish defaults in certain fields in the reports and application screens, to identify the installation, and to tell in which directory ***MCHAS*** is installed.

How to Update Installation Information

1 From the **System Administration** Application Screen, click on the **Installation Info** button.

The Installation Information Application Screen appears (Figure 1).

Note: The Installation Directory must be kept current as this information is used throughout the *MCHAS* system. DO NOT change unless your Network or System Administrator instructs you to do so because of a change in your MCHAS installation.

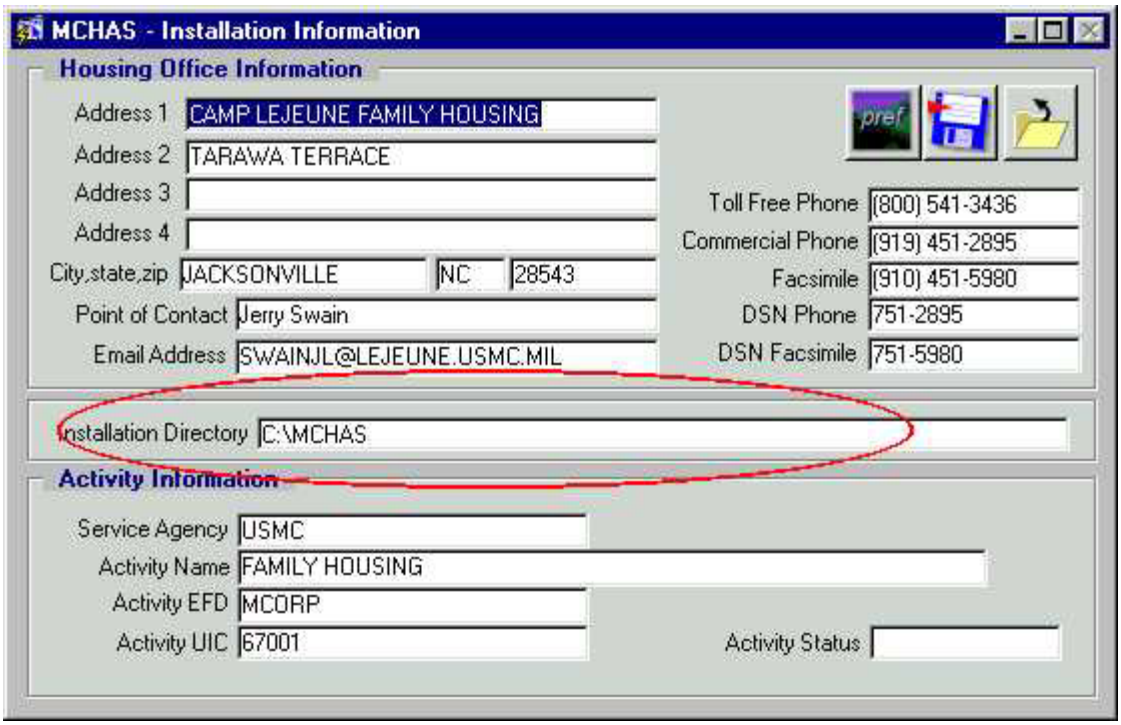


Figure 1 - Installation Information

2 Click in the fields where changes or additions need to take place.

3 Click on the **Save** button to save changes.

4 Click on the **Close** button to close the application screen.

Setting Preferences

You must have Administrative Privileges to set Preferences.

1. From the **Main Screen** click on **System Admin**.
2. Click on the **Installation Info** button.
3. Click on the **Preferences** button.
4. The Preferences screen will appear (Figure 2).
5. Fill in the default values for your installation. Set your Inspection Scheduling defaults and click the **Apply** button. (You will be asked to confirm changing the Timeline Interval, answer Yes).
6. If you want to confirm removing Sponsors from the Waitlist upon assignment to quarters, select the checkbox.
7. Click the **Close** button.

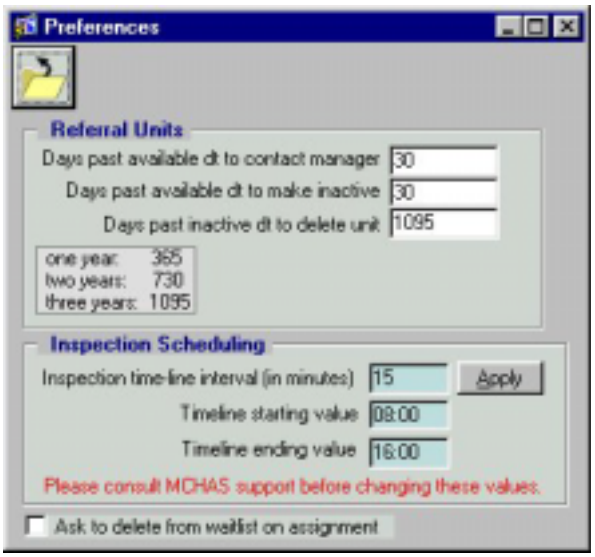


Figure 2 - Setting Preferences

Setting Referral Units Dates sets the defaults for Contact Date, Inactive Date and Delete Date for Referral Units. When you enter the Available Date for a Referral and press the button for date calculations, those calculations are based upon the numbers here.

Inspection scheduling numbers affect the Inspection Timeline appearance. If your inspection day is from 07:00 to 14:00 and those numbers are entered here, the timeline will start at 7am and end at 2pm. Intervals (15, 30 & 60) are the number of minutes for each increment within the hours of the inspection day as shown on the Timeline.

Security Setup

MCHAS Security consists of four major areas:

- ✓ Application
- ✓ Groups
- ✓ Users
- ✓ Components

The Application

Contains the installation information for MCHAS (Figure 3).

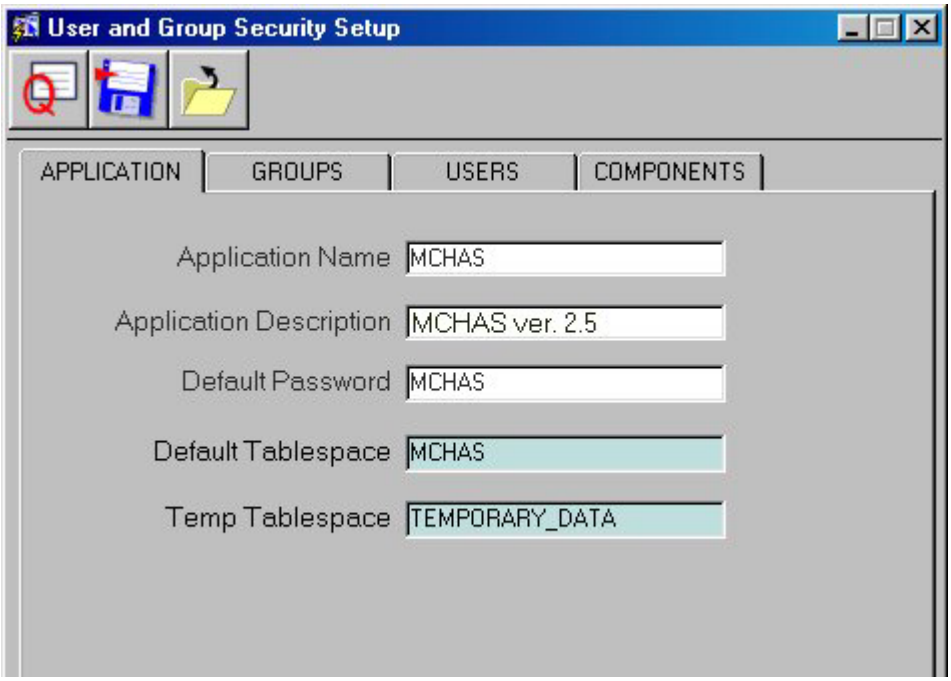


Figure 3 - Application Tab

Groups - Modifying and Creating

Groups are MCHAS' way of controlling access to the Oracle database tables and the permissions for each. Groups are created first, the table settings are made and then Users are placed into the Groups.

1. From the **User and Group Security Setup** screen, click on the **GROUPS** tab.
2. **Double-click** in the Group Name field to select a specific group and click **OK.**
3. Press the **Query** button to execute the query.

   You will see the Group Name, each of the tables in MCHAS and the permissions that group has for each of the tables (Figure 5).

Figure 4 - Create New Group

4. To create a new Group, click the **Insert a Record** button.
5. A popup appears asking for a name for the new Group, 8-character maximum (Figure 4). Enter the new Group name and press the **Save** button.
6. **Double-click** in the Group Name field, select the newly created Group name and click **OK**.
7. Press the **Query** button to execute the query.
8. You can see the newly created group and its permissions. Modify if necessary.

Group Name	Table Name	Select	Insert	Update	Delete
DEFAULT	APPLICATIONS	Y	Y	Y	Y
DEFAULT	ASSIGN_HISTORY	Y	Y	Y	Y
DEFAULT	BAD_BOYS	Y	Y	Y	Y
DEFAULT	BAH_RATES	Y	Y	Y	Y
DEFAULT	CHARGE	Y	Y	Y	Y
DEFAULT	CHARGE_SCHEDULE	Y	Y	Y	Y
DEFAULT	COMPONENTS	Y	Y	Y	Y
DEFAULT	CQ_OCCUPANT	Y	Y	Y	Y
DEFAULT	DEFAULT_CHARGES	Y	Y	Y	Y
DEFAULT	DEPENDENT	Y	Y	Y	Y
DEFAULT	FIELD_DEFAULTS	Y	Y	Y	Y
DEFAULT	GOX	Y	Y	Y	Y
DEFAULT	GROUPTABS	Y	Y	Y	Y
DEFAULT	HOUSING_METERS	Y	Y	Y	Y
DEFAULT	HOUSING_UNIT	Y	Y	Y	Y
DEFAULT	HOUSING_UNIT_GRADE	Y	Y	Y	Y
DEFAULT	INSTALLATION_INFO	Y	Y	Y	Y
DEFAULT	METER_TRANS	Y	Y	Y	Y
DEFAULT	MILITARY_SPOUSE	Y	Y	Y	Y
DEFAULT	MODULES	Y	Y	Y	Y
DEFAULT	MONTHLY_PAYMENT_FLAG	Y	Y	Y	Y

Figure 5 - Groups Tab

Consult your DBA or System Administrator before changing any of these settings if you are unsure of what permissions should be granted.

Users - Viewing and Creating

This is the section of Security in which the Users for the application MCHAS are set up.

1. From the **User and Group Security Setup** screen, click on the **USERS** tab.
2. Press the **Query** button to view all Users.

   You will see the User ID, Group Name, and the complete Name of your Users (Figure 7).
3. To create a new User, click the **Insert a Record** button.
4. The Create New User input screen appears (Figure 6).
5. **Double-click** in the Group Name field to access the Groups already created. Select from the list and click **OK**.
6. Click on the **Save** button.
7. You will receive a message when the User has been created, click **OK** (Figure 8).

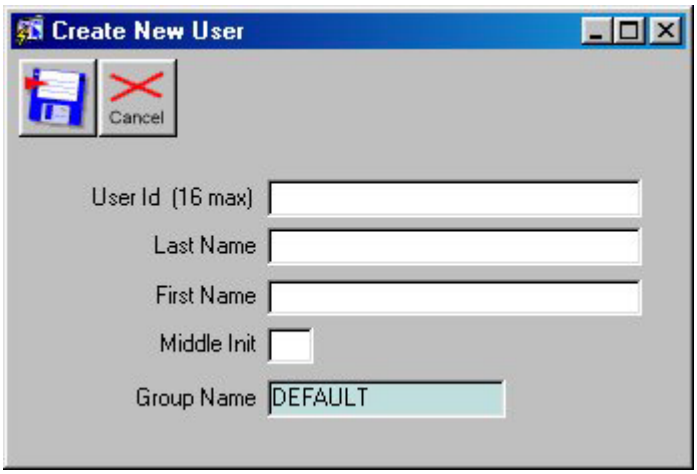


Figure 6 - Create New User

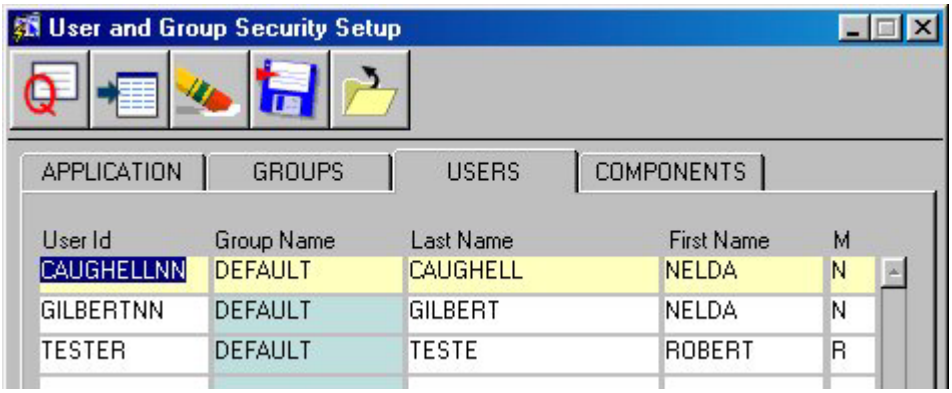


Figure 7 - Users Tab

Figure 8 - User Created

Modifying Users

If you find that an error was made during the creation of a user, **DO NOT** modify the Users information on the **Users** tab. Since users are created at the database, they must be DROPPED and recreated.

8. From the **User and Group Security Setup** screen, click on the **USERS** tab.
9. Press the **Query** button to view all Users or enter the User ID, Last Name or First Name for the User you wish to drop and press the **Query** button.
10. With the User selected (the row will be yellow) click the **Delete Current User** button.
11. You will receive two prompts to confirm the delete (Figure 9 and Figure 10). If you wish to continue, click **Yes** for both prompts.

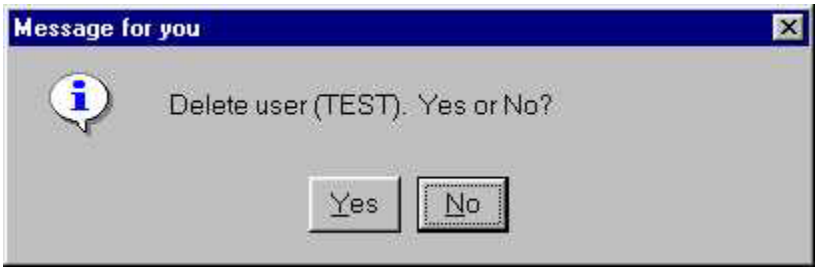


Figure 9 - Delete user prompt 1

Figure 10 - Delete user prompt 2

When the user has been dropped from the database, you will receive the confirmation on the status bar (Figure 11).

Figure 11 - Status bar confirmation for save

12. Repeat the steps for creating a user above (Users - Viewing and Creating) to re-create this User.

About Components

The Components tab of the Security Setup is where User access to the individual screens, buttons and functions within MCHAS are granted to the User. Since access to certain Modules is restricted, like the System Admin. Module and the Financial Module, access is granted to the User at the component level.

1. From the **User and Group Security Setup** screen, click on the **COMPONENTS** tab.

   Pressing the **Query** button will show all Users and the components that have been granted to them (Figure 13).

2. **Double-click** in the User ID field to access the list of Users created, select one from the list and click **OK**.
3. Press the **Query** button to view only *that* user's components.
4. To grant components to a new User or modify for an existing user, click the Click the **Quick Setup** button.

5. **Double-click** in the User ID field (Figure 12) and choose the User from the list and click **OK**.

Figure 12 - Component Quick Setup

6. For Starters, select the checkbox for **Standard components**.
7. Select any other components that the User will need to perform their job.
8. Click the **Save** button, then **Close**.

Components have been divided into the Modules of MCHAS. By clicking the Referral components checkbox, the User will receive access to all components in the Referral Module. By using this type of grouping, the System Administrator does not need to remember which components are for the Referral Module the User receives them all.

If you need to modify someone's components, remember when you use the Quick Component Setup it is like starting from scratch each time. If a User previously had only Standard components and you wish to add Base Housing components, when you set up the components, you **must** select Standard components again as well as the new Base Housing components.

Standard Components for the General User are: Standard, Base Housing and Wait List.

Refer to your DBA or System Administrator for a complete list or components and modules to guide you through setting up user components.

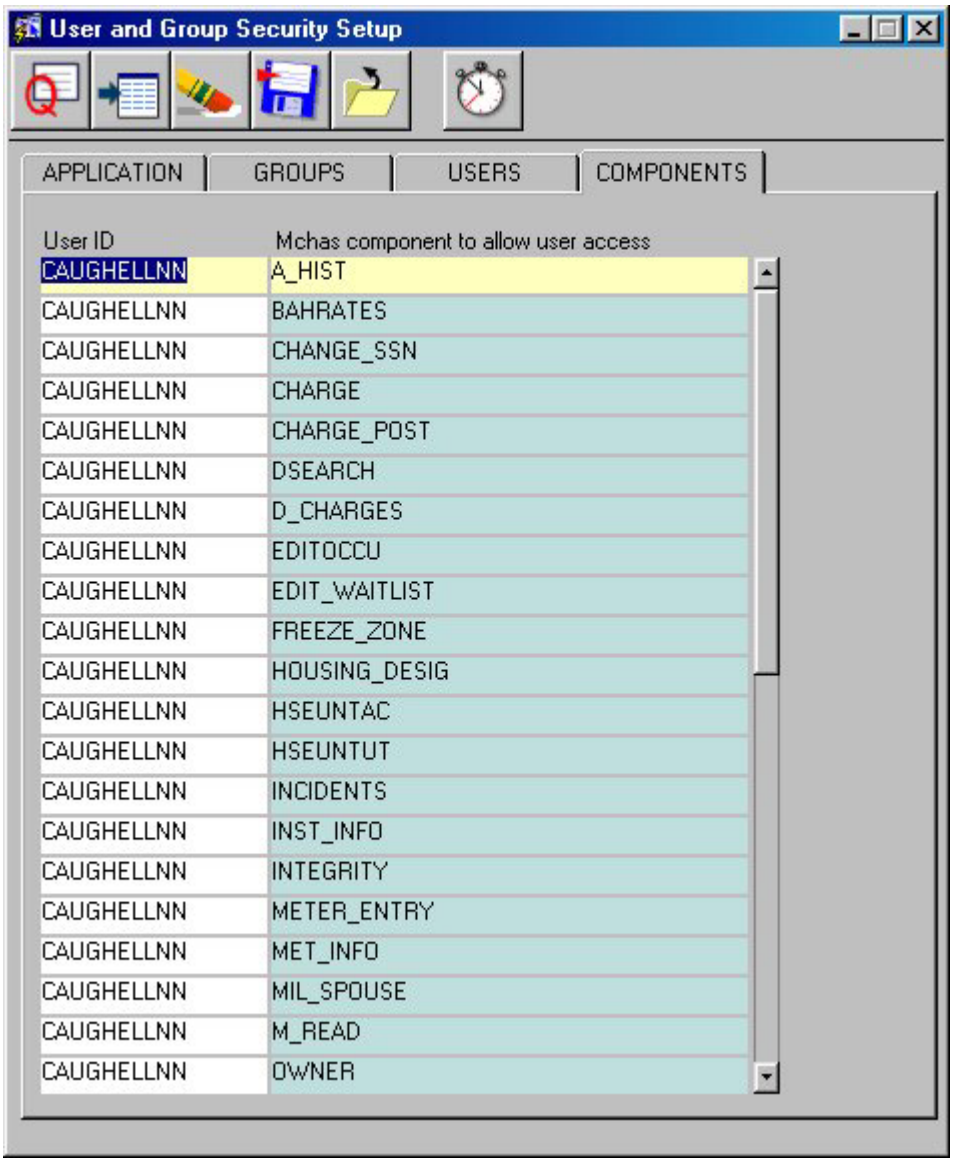


Figure 13 - Components Tab

Adding SQR Reports

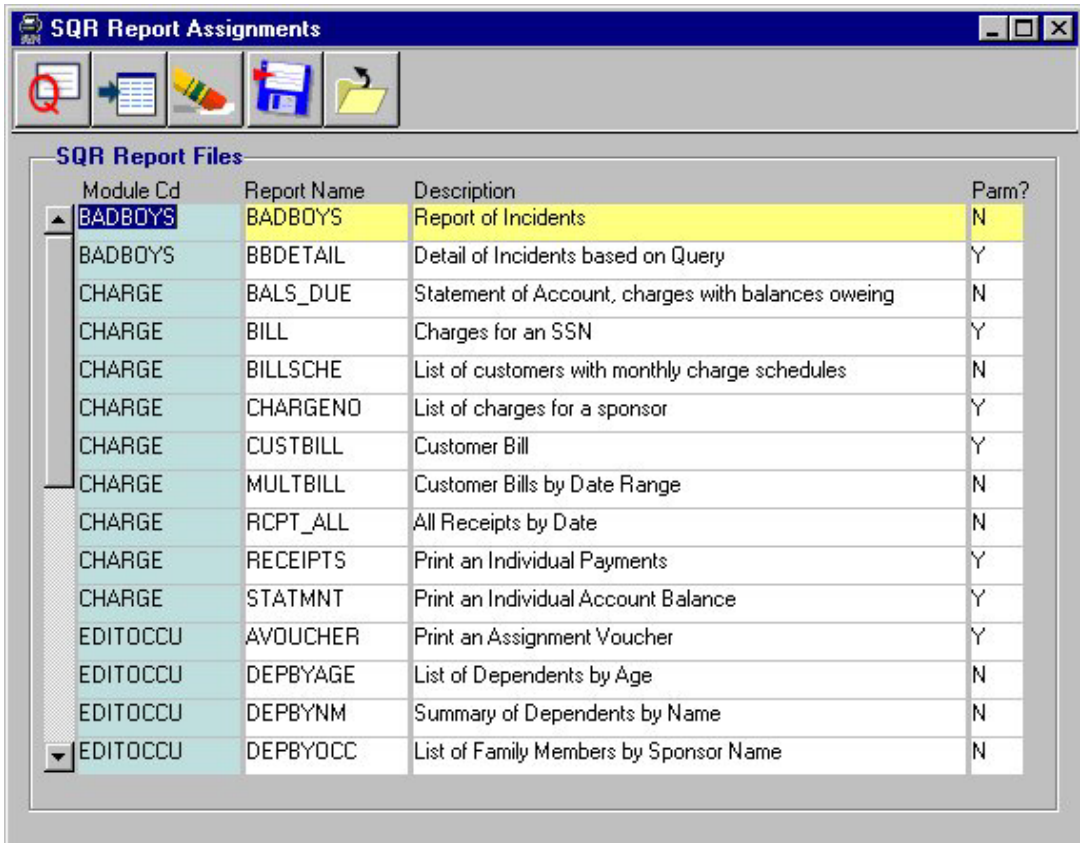
1. From the **System Administration Application Screen**, click on **Add SQR Reports**.
2. The **SQR Report Assignments Application Screen** appears (Figure 14)
3. Input the desired module or **double-click** in the Module Name field and make a selection from the value list.
4. Click on the **Query** button to execute the query.
5. Click on the **Insert a Record** button to insert a new row.
6. Input the report name.
7. Tab to the Description field and input the Description of the new report.
8. Tab to the Parm? Field, input a Y for Yes or N for No.

9. Click on the **Save** button to save changes.
10. Click on the **Close** button to close the application screen.

Note: Once the insert button is activated, information must be entered in order to exit the screen. The record can then be removed by clicking the **Delete a Record**.

Table 0.1 - SQR Reports Field Definitions

Input Field	Description
Module Cd	8 character name for the Module the report is to be attached to
Report Name	Report names utilize the 8.3 naming convention. When adding your own custom reports, make sure the report name does not exceed the 8 character limit
Description	Brief but meaningful description of the report
Parm?	Parameter - Does the report require a parameter from MCHAS to run.



Module Cd	Report Name	Description	Parm?
BADBOYS	BADBOYS	Report of Incidents	N
BADBOYS	BBDETAIL	Detail of Incidents based on Query	Y
CHARGE	BALS_DUE	Statement of Account, charges with balances oweing	N
CHARGE	BILL	Charges for an SSN	Y
CHARGE	BILLSCHE	List of customers with monthly charge schedules	N
CHARGE	CHARGENO	List of charges for a sponsor	Y
CHARGE	CUSTBILL	Customer Bill	Y
CHARGE	MULTBILL	Customer Bills by Date Range	N
CHARGE	RCPT_ALL	All Receipts by Date	N
CHARGE	RECEIPTS	Print an Individual Payments	Y
CHARGE	STATMNT	Print an Individual Account Balance	Y
EDITOCCU	AVOUCHER	Print an Assignment Voucher	Y
EDITOCCU	DEPBYAGE	List of Dependents by Age	N
EDITOCCU	DEPBYNM	Summary of Dependents by Name	N
EDITOCCU	DEPBYOCC	List of Family Members by Sponsor Name	N

Figure 14 - SQR Report Assignments

Setting User Passwords

1. If you need to reset the password for a User, click the **Set User Passwords** button.
2. Enter the User Login ID.
3. Enter the new Password, confirm the password by entering it again
4. Click the **Save** button.

Data Cleanup

1. From the **System Administration Application Screen**, click on the **Data Cleanup** button.
2. The Data Cleanup Screen appears (Figure 15).
3. Click on the button for the Cleanup procedure to start the Record Integrity.
4. Click on the **Close** button to close the application screen.

Note: This process can take a long time. Once begun, it should not be interrupted.

Figure 15 - Data Cleanup

Value Tables

MCHAS allows for several site-specific tables to be updated by the *MCHAS* Administrators. If the current user has permission, the System Administration Application Screen appears (Figure 16).

1 From the **Main Menu** screen, click on the **System Admin Module**.

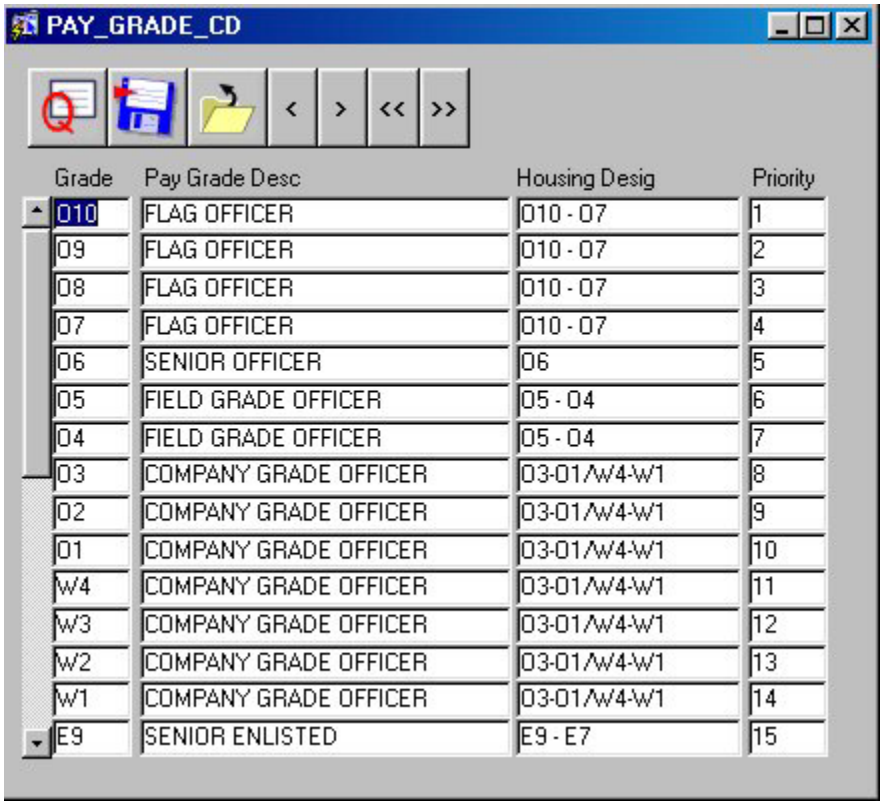
A Selection List will pop up.

2 Select **SYSTEM ADMIN** from the Pop-up menu and click **OK**.

Figure 16 - System Administration

Pay Grades

The Pay Grade table in the database should contain ONLY VALID Marine Corps pay grades. This table maintains information about the houses a certain pay grade may occupy, through the housing designator, and the priority the pay grade has in the Waitlist.



Grade	Pay Grade Desc	Housing Desig	Priority
010	FLAG OFFICER	010 - 07	1
09	FLAG OFFICER	010 - 07	2
08	FLAG OFFICER	010 - 07	3
07	FLAG OFFICER	010 - 07	4
06	SENIOR OFFICER	06	5
05	FIELD GRADE OFFICER	05 - 04	6
04	FIELD GRADE OFFICER	05 - 04	7
03	COMPANY GRADE OFFICER	03-01/W4-W1	8
02	COMPANY GRADE OFFICER	03-01/W4-W1	9
01	COMPANY GRADE OFFICER	03-01/W4-W1	10
W4	COMPANY GRADE OFFICER	03-01/W4-W1	11
W3	COMPANY GRADE OFFICER	03-01/W4-W1	12
W2	COMPANY GRADE OFFICER	03-01/W4-W1	13
W1	COMPANY GRADE OFFICER	03-01/W4-W1	14
E9	SENIOR ENLISTED	E9 - E7	15

Figure 17 - Pay Grades

Table 0.2 - Pay Grade Codes Description

Input Field	Description
Grade	Valid Pay Grades
Pay Grade Desc.	Description for the Pay Grade
Housing Desig.	Housing Designator that tells MCHAS which houses this Pay Grade can occupy
Priority	The Priority to which the Pay Grade is given on the Wait List

To add a new record,

1 Using the scroll bar, scroll down and place the cursor in the last record of the list then press the down arrow.

The cursor will appear on an empty row.

2 Input the new record.

3 Click on the **Save** button to save changes.

4 Click on the **Close** button to close the application screen.

Zip Codes

The Zip table contains local area zip codes and is used to create the List of Values for Referral Housing.

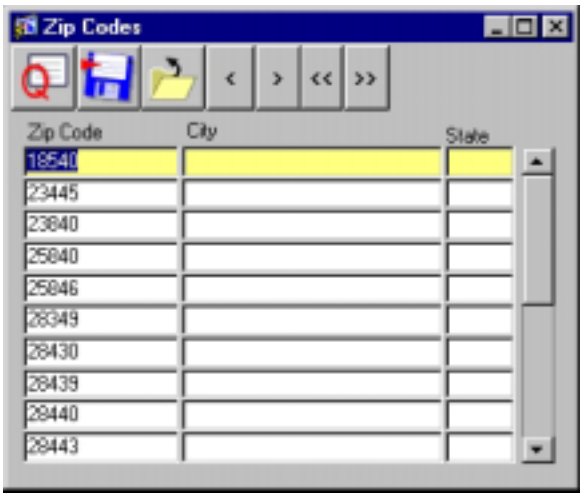


Figure 18 - Zip Table

Value Lists

1. From the **System Administration Application Screen**, click on the **Value List** button.

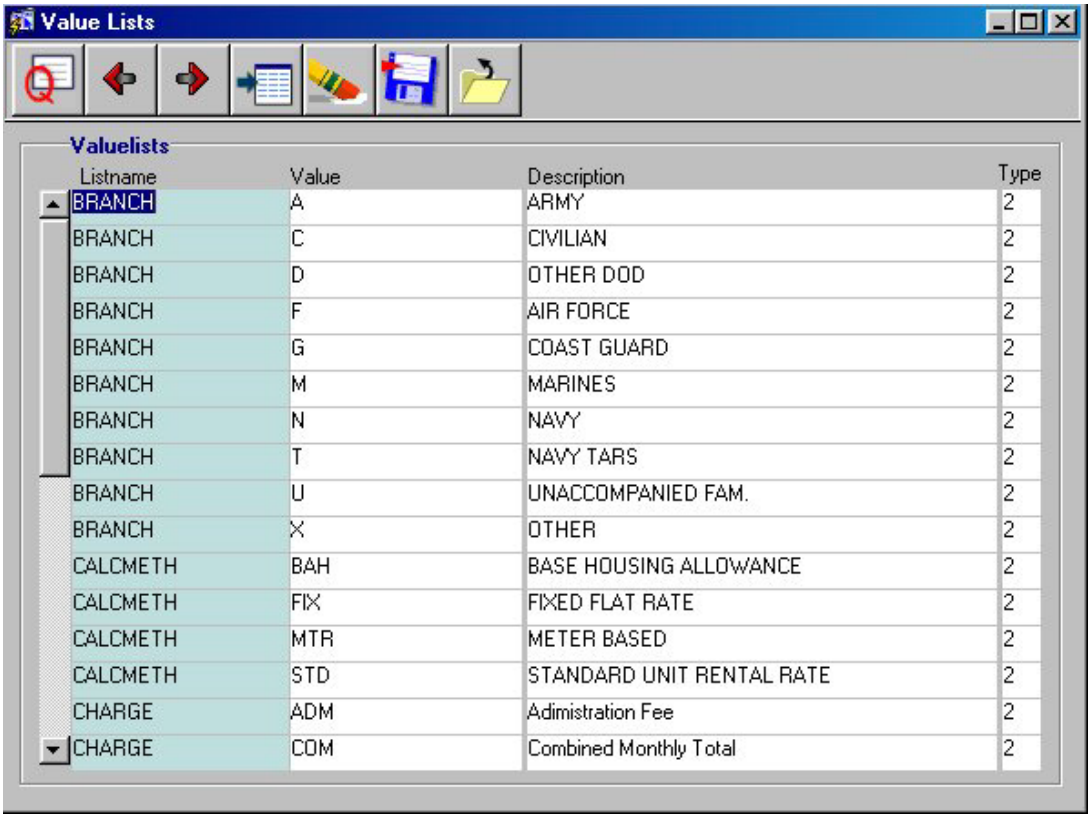


Figure 19 - Valuelists

2. The value list application screen opens (Figure 19) showing all lists.

3. To view only one list, press the **Query** button and input the desired value list or **double-click** in the Listname field and make a selection from the value list.
4. Click on the **Query** button *again* to execute the query.
5. Click in the field you would like to update. Make your changes.
6. Click on the **Save** button to save the changes.
7. To add a new record to the value list, place the cursor in a field.
8. Click on the **Insert a Record** button.

   MCHAS has inserted a blank row to accept the new record.
9. Input the desired data, repeating the Listname exactly.
10. Click on the **Save** button to save changes.
11. Click on the **Close** button to close the application screen.

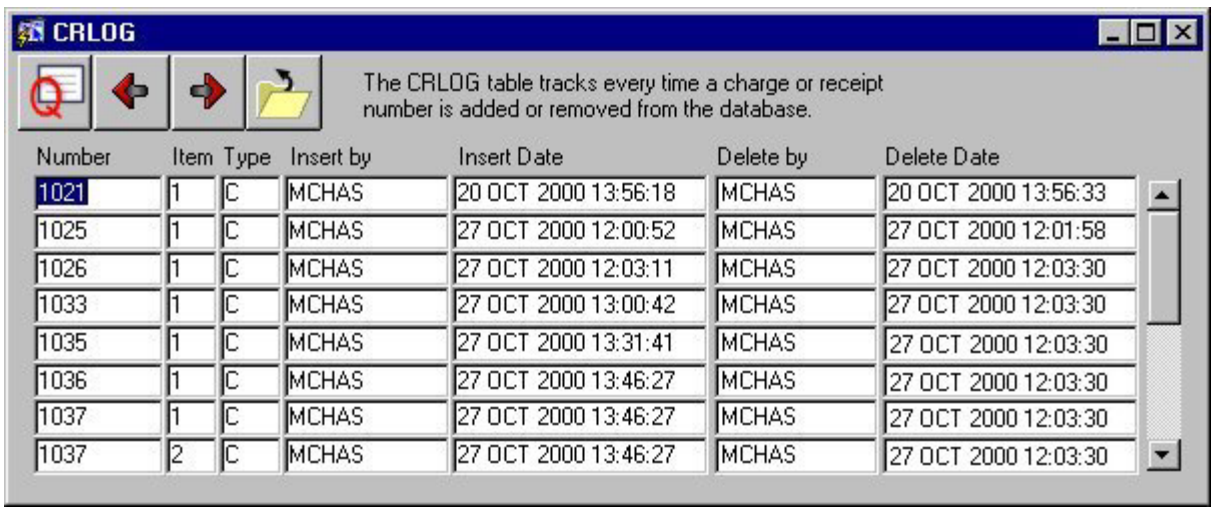
Note: In the Type field, input a 2 (for values with descriptions) and 1 (for values that do not have a description).

Table 0.3 - Value List Definitions

Input Field	Description
Listname	The Name given to the Valuelist
Value	The Value that MCHAS will use
Description	A Description for the Value
Type	1 = No Description 2 = Description

The CR Log

1. From the **Main Menu**, click on **System Admin.**
2. From the Selection List, choose **CRLOG** and click **OK**.
3. The CR Log screen opens showing inserts and deletes from the Charge and Receipt screens.



Number	Item	Type	Insert by	Insert Date	Delete by	Delete Date
1021	1	C	MCHAS	20 OCT 2000 13:56:18	MCHAS	20 OCT 2000 13:56:33
1025	1	C	MCHAS	27 OCT 2000 12:00:52	MCHAS	27 OCT 2000 12:01:58
1026	1	C	MCHAS	27 OCT 2000 12:03:11	MCHAS	27 OCT 2000 12:03:30
1033	1	C	MCHAS	27 OCT 2000 13:00:42	MCHAS	27 OCT 2000 12:03:30
1035	1	C	MCHAS	27 OCT 2000 13:31:41	MCHAS	27 OCT 2000 12:03:30
1036	1	C	MCHAS	27 OCT 2000 13:46:27	MCHAS	27 OCT 2000 12:03:30
1037	1	C	MCHAS	27 OCT 2000 13:46:27	MCHAS	27 OCT 2000 12:03:30
1037	2	C	MCHAS	27 OCT 2000 13:46:27	MCHAS	27 OCT 2000 12:03:30

Figure 20 - CRLOG

You may Query the CR Log by Charge/Receipt Number, Item Number, Type (C=Charge, R=Receipt), Insert by or Delete by.

4. To Query the CRLOG, press the **Query** button, enter the information you want to query on and press the **Query** button again. Those records meeting the query criteria are returned.
5. Click on the **Close** button to close the application screen.

MCHAS Data Dictionary

A Data Dictionary contains all information about your database tables. It provides you with the table name, column name, data type, data length and whether NULLS are allowed.

ALL_TAB_COLUMNS

ALL_TAB_COLUMNS contains information about all the tables in the database.

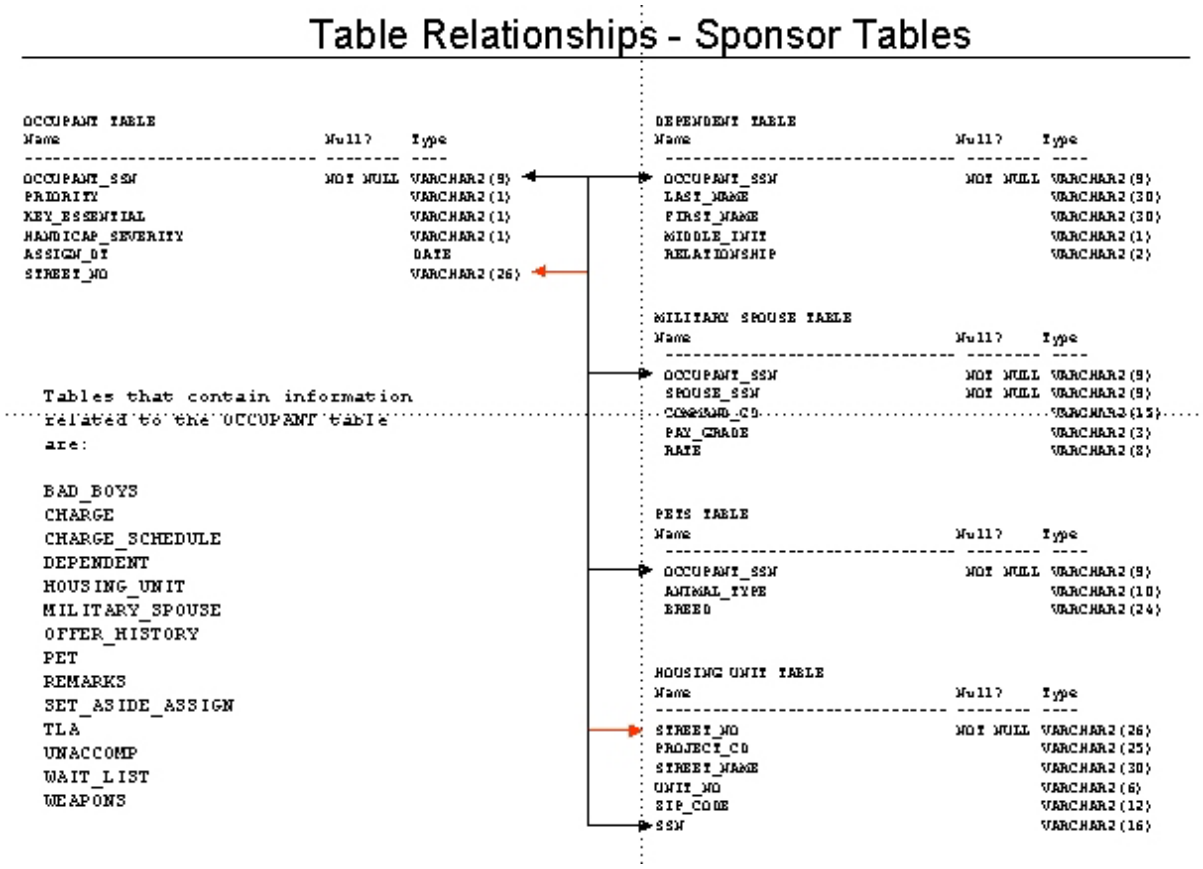
```
SQL> desc all_tab_columns
 Name                            Null?    Type
 ------------------------------- -------- ----
 OWNER                           NOT NULL VARCHAR2(30)
 TABLE_NAME                      NOT NULL VARCHAR2(30)
 COLUMN_NAME                     NOT NULL VARCHAR2(30)
 DATA_TYPE                                VARCHAR2(9)
 DATA_LENGTH                     NOT NULL NUMBER
 DATA_PRECISION                           NUMBER
 DATA_SCALE                               NUMBER
 NULLABLE                                 VARCHAR2(1)
 COLUMN_ID                       NOT NULL NUMBER
 DEFAULT_LENGTH                           NUMBER
 DATA_DEFAULT                             LONG
 NUM_DISTINCT                             NUMBER
 LOW_VALUE                                RAW(32)
 HIGH_VALUE                               RAW(32)
 DENSITY                                  NUMBER
 NUM_NULLS                                NUMBER
 NUM_BUCKETS                              NUMBER
 LAST_ANALYZED                            DATE
 SAMPLE_SIZE                              NUMBER
```

SQR DATADICT Report

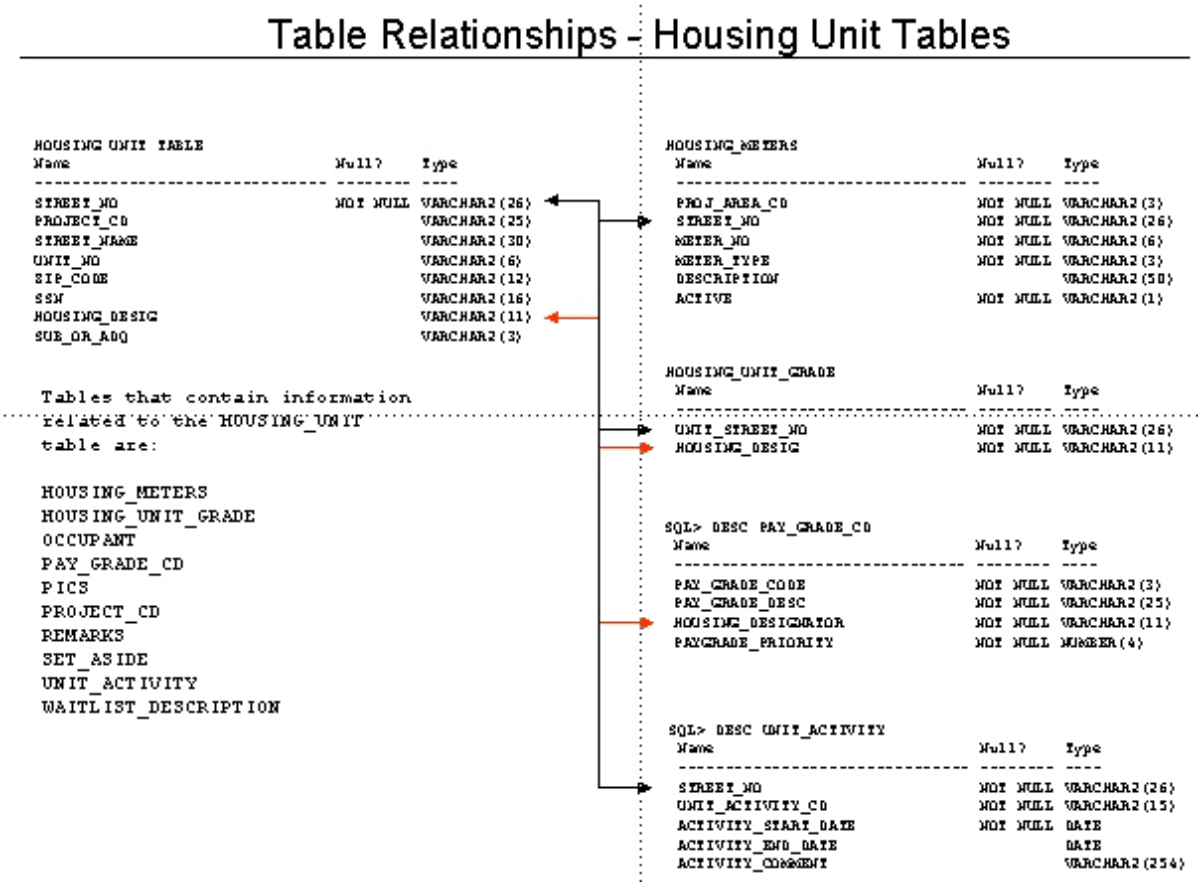
The SQR Report MCHASDIC.SQT, which is installed in the Administration Module, provides information about the MCHAS tables and can be printed at any time. If a change is made to the database, print again to update to the most current table structure.

MCHAS Table Relationships

Sponsor Tables



Housing Unit Tables



Wait List Tables

Table Relationships - Wait List Tables

```
SQL> DESC WAIT_LIST
 Name                            Null?    Type
 ------------------------------- -------- ----
 WAITLIST_CD                     NOT NULL VARCHAR2(10)
 ACTIVE                                   VARCHAR2(1)
 FREEZE_SOME                              VARCHAR2(1)
 FREEZE_DT                                DATE
 PRIORITY                                 VARCHAR2(1)
 CONTROL_DT                               DATE
 PAYGRADE_PRIORITY                        NUMBER(4)
 RANK_DT                                  DATE
 APP_DT                                   DATE
 POS                                      NUMBER(22)
 OCCUPANT_SSN                    NOT NULL VARCHAR2(9)
 LAST_NAME                                VARCHAR2(30)
 FIRST_NAME                               VARCHAR2(15)
 MIDDLE_INIT                              VARCHAR2(1)
 PAY_GRADE                                VARCHAR2(3)
 HOME_PHONE                               VARCHAR2(24)
 WORK_PHONE                               VARCHAR2(24)
 COMMAND_CD                               VARCHAR2(15)
 PROJ_AREA_CD                             VARCHAR2(3)
 NO_BEDROOMS                              VARCHAR2(1)
 HOUSE_DES                                VARCHAR2(11)
 SET_ACTIVE_DT                            DATE
 INSERT_BY                                VARCHAR2(32)
 INSERT_DT                                DATE
 UPDATE_BY                                VARCHAR2(32)
 UPDATE_DT                                DATE
```

Tables that contain information related to the WAIT_LIST table are:

OCCUPANT
PAY_GRADE_CD
WAITLIST_CD
WAITLIST_DESCRIPTION

```
OCCUPANT TABLE
 Name                            Null?    Type
 ------------------------------- -------- ----
 OCCUPANT_SSN                    NOT NULL VARCHAR2(9)
 PRIORITY                                 VARCHAR2(1)
 KEY_ESSENTIAL                            VARCHAR2(1)
 HANDICAP_SEVERITY                        VARCHAR2(1)
 ASSIGN_DT                                DATE
 STREET_NO                                VARCHAR2(26)
```

```
SQL> DESC PAY_GRADE_CD
 Name                            Null?    Type
 ------------------------------- -------- ----
 PAY_GRADE_CODE                  NOT NULL VARCHAR2(3)
 PAY_GRADE_DESC                  NOT NULL VARCHAR2(25)
 HOUSING_DESIGNATOR              NOT NULL VARCHAR2(11)
 PAYGRADE_PRIORITY               NOT NULL NUMBER(4)
```

```
SQL> DESC WAITLIST_CD
 Name                            Null?    Type
 ------------------------------- -------- ----
 WAITLIST_CD                     NOT NULL VARCHAR2(10)
 WAITLIST_DESC                   NOT NULL VARCHAR2(254)
 SUB_OR_ADQ                               VARCHAR2(3)
```

```
SQL> DESC WAITLIST_DESCRIPTION
 Name                            Null?    Type
 ------------------------------- -------- ----
 WAITLIST_CD                     NOT NULL VARCHAR2(10)
 NO_BEDROOMS                     NOT NULL VARCHAR2(1)
 HOUSE_DES                       NOT NULL VARCHAR2(11)
 PAY_GRADE                       NOT NULL VARCHAR2(3)
 PROJ_AREA_CD                    NOT NULL VARCHAR2(3)
```

Notes

SQR Reports

Introduction to SQR Reports

SQR[1] report writing is a comprehensive multi-RDBMS[2] reporting system designed to deliver a full range of SQL[3] reporting requirements.

SQR supports the full complement of SQL commands. Using SQR, you can build and print complex reports that execute SQL statements against your database.

Your SQR Report is divided into three sections (Figure 1):

1. The **Heading**, that portion of the report along the top of the page that will repeat over all pages. This may include column headings, a descriptive report title, etc.
2. The **Body**, that portion of the report that returns the results of the Query.
3. The **Footing**, that portion of the report along the bottom of the page that will repeat over all pages. This may include page numbers, report names, etc.

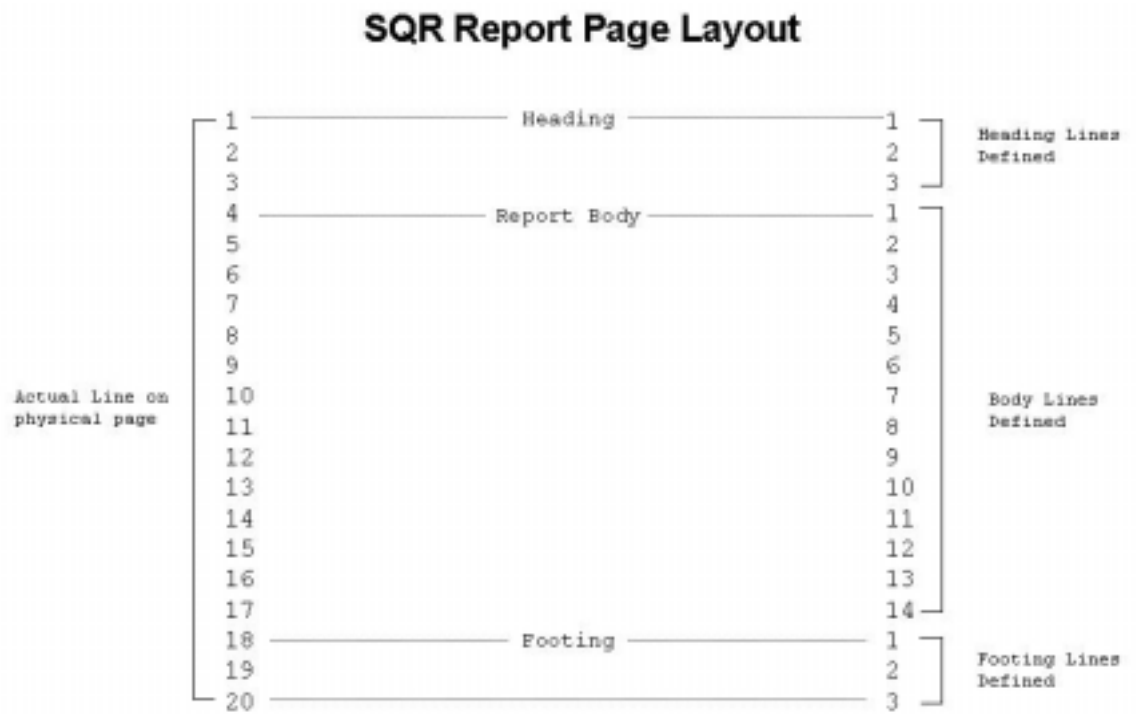


Figure 1 - SQR Report Page Layout

[1] Structured Query Report
[2] Relational Database Management System
[3] Structured Query Language

In SQR, you must place each command on a new line. You may indent SQR commands.

An output page can be thought of as a grid of lines and columns. The "(1,1)" indicates line one, column one, which is the top left corner of the page. The numbers in parentheses give the position for printing. A position in SQR is expressed as three numbers in parentheses (line,column,width) where line is the line number, column is the column (character position) and width is the width of the text.

SQR Data Elements

SQR data elements include columns, variables, and literals. Each begins with a special character which denotes the type of data element.

- Columns are fields defined in the database.

  & begins a database column or expression name. It can be any type (for example, character, number, or date) and, except for dynamic columns and database or aggregate functions, is declared automatically for columns defined in a query.

- Variables are storage places for text or numbers that you define and manipulate.

  $ begins a text or date variable.
  # begins a numeric variable.
  @ begins a variable name for a marker location. Marker locations are used to identify positions to begin for printing inside a BEGIN-DOCUMENT paragraph.

- Literals are text or numeric constants.

  A single quote begins and ends a text literal. For example, 'Hello'.
  0-9 begin any numeric literals. Numerals that include digits with an optional decimal point and leading sign are acceptable numeric literals. For example, -543.21. Numeric literals can also be expressed in scientific form. For example, 1.2E5.

Connecting to a Database

SQR Reports may be written in any text editor, or you may use the SQR Workbench program to write your reports.

4. Open the SQR Workbench Program. From the Start Menu, select Programs > SQRIBE Products > SQR WorkBench.

5. From the File Menu, Select Logon to Database (Figure 2).

6. Under the Machine Data Source tab, find the ODBC connection for your database (Figure 3).

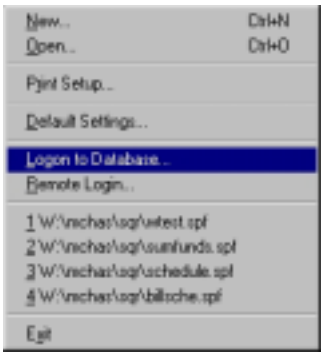


Figure 2 - Logon to Database

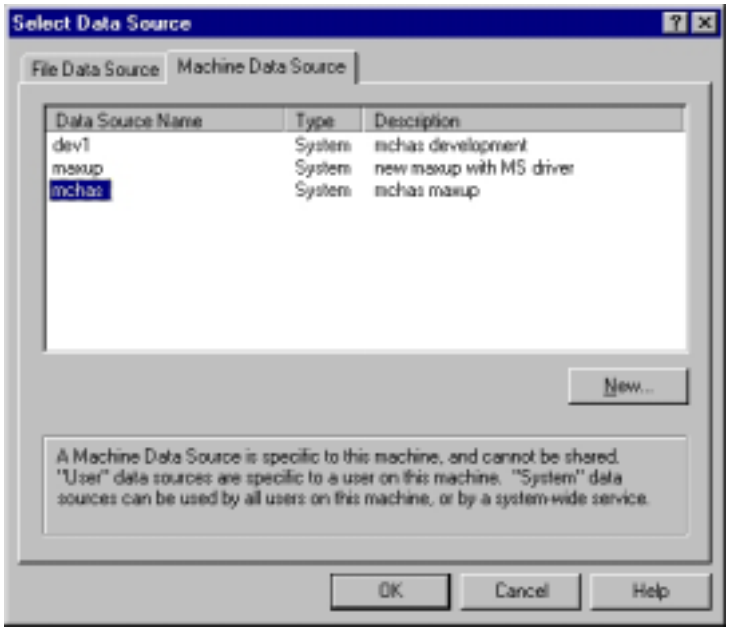


Figure 3 - Select Data Source

7. When prompted, Logon to the database (Figure 4). SQR WorkBench will open to a blank window.
8. From the File Menu, select New (Figure 5).
9. You are presented with the many types of reports you can create (Figure 6). For purposes of this workshop, click on the Editor button.

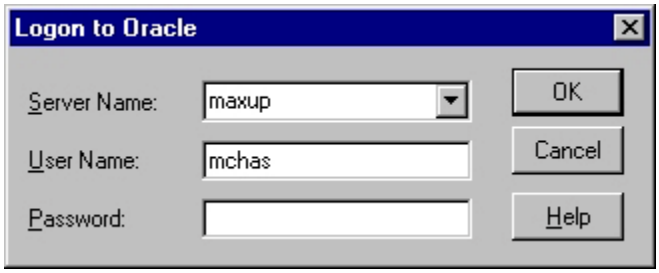


Figure 4 - Logon to Oracle

Figure 5 - Create a New Report

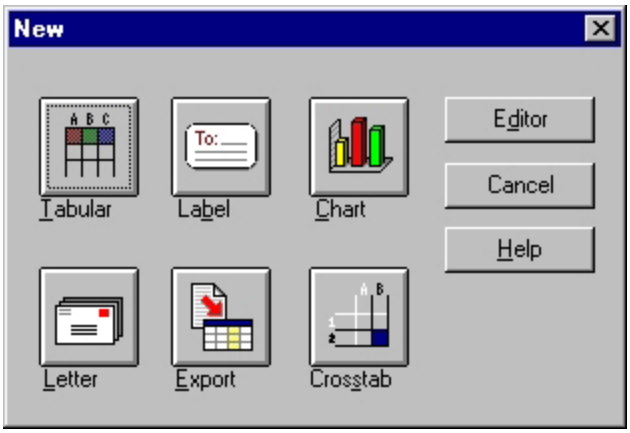


Figure 6 - New Report Options

General SQR Report Sections

Your SQR report consists of the following Sections:

1. SETUP section - REQUIRED
2. HEADING section
3. FOOTING section
4. REPORT section - REQUIRED
5. PROCEDURE section - REQUIRED
   A. SELECT paragraphs
   B. SQL paragraphs
   C. DOCUMENT paragraphs

SETUP Section

Use the SETUP section to declare overall characteristics of the report. Within the Setup section, a sub-section "declare-layout" sets up the page size and orientation, similar to page setup in a word processor.

```
begin-setup
declare-layout default
    paper-size=(8.5,11)  ! The page is 8.5 inches wide, 11 long
    left-margin=0.5
    orientation=portrait
end-declare
end-setup
```

Also use the SETUP section to define column positions.

```
begin-setup
declare-layout default
    paper-size=(8.5,11)              !8.5 inches wide, 11 long
    left-margin=0.5
    orientation=portrait
```

```
end-declare
    #define col_proj          1     !Define the Column & position
    #define col_addr         10     !Define the Column & position
    #define col_beds         25     !Define the Column & position
    #define col_bath         30     !Define the Column & position
    #define col_sq_ft        35     !Define the Column & position
    #define col_desig        40     !Define the Column & position
    #define col_act          45     !Define the Column & position
    #define col_sub          50     !Define the Column & position
    #define col_vac          55     !Define the Column & position
end-setup
```

HEADING Section

Use to define the heading for the report. Similar to "Headers" in a word processor, the heading repeats across all pages of the report. The number following "begin-heading" tells SQR how many lines to allow for the heading.

```
begin-heading 3                        !#lines for Heading
alter-printer point-size=8 font=4      !set the font
print 'Family Housing Report' (1,1)    !Place line 1, position 1
print 'Second Line'           (2,1)    !Place line 2, position 1
print 'Last Line'             (3,1)    !Place line 3, position 1
graphic (,1,73) horz-line 10           !Place a line
end-heading
```

FOOTING Section

Use to define the footing or "footer" of the report. As with the Heading section, the footing repeats across all pages of the report and the number of lines allowed for the footing are declared in the same manner.

```
begin-footing 3                        !#lines for Footing
graphic (1,1,75) horz-line 10          !Place a line
    alter-printer point-size=8 font=4  !set the font
date-time                     (2,1)    !Place line 1, position 1
page-number                   (2,15)   !Place line 2, position 15
print   'report_name.SQT'     (3,50)   !Place line 3, position 50
end-footing
```

REPORT Section

The Report section is REQUIRED for SQR reports. From this section calls are made to the select paragraphs in the Procedure section.

```
begin-program
    do main_select                    !Call the SELECT paragraph
end-program
```

or

```
begin-report
    do main_select                    !Call the SELECT paragraph
end-report
```

PROCEDURE Section

There are three types of paragraphs in the Procedure section: SELECT, SQL and DOCUMENT.

SELECT paragraphs

Select paragraphs make up the query to the database for the data that you want the report to return.

```
begin-procedure main_select        !Name the Procedure
let #count = 0                     !Set the count to 0
begin-select                       !Begin the select statement
project_cd            &project     !Database column and SQR name
   add 1 to #count                 !Advance the count by 1
actual_street_no      &street_no   !Database column and SQR name
street_name           &st_name     !Database column and SQR name
no_bedrooms           &beds        !Database column and SQR name
housing_desig         &desig       !Database column and SQR name
active_status         &active      !Database column and SQR name
sub_or_adq            &sub         !Database column and SQR name
vacate_dt             &vacate_dt   !Database column and SQR name
   do print_list                   !Call to another procedure
from housing_unit                  !Database table
where project_cd = 'ASO'           !Qualify the select
order by project_cd                !Order by
end-select                         !Close the Query
print 'Housing Units' (+2,1) bold
print #count          ( ,25) bold edit 9,999
end-procedure                      !End the Procedure
```

Using Column Variables

If you want to use the value of a column for another purpose, create a Column variable, preceded with an ampersand (&).

```
begin-select
phone &phone
position (+1)
from customers
end-select
```

Note that the column variable &phone inherited its name from the phone column. This is the default, but you can change it, as the following example demonstrates.

```
begin-select
phone &cust_phone
position (+1)
from customers
end-select
```

SQL paragraphs

Although SELECT may be the most common SQL statement, you can also perform other SQL commands in SQR. To perform a SQL statement other than a SELECT statement you must use the BEGIN-SQL paragraph.

```
begin-procedure set_nls_date_format
begin-sql
     alter session set nls_date_format = 'DD-MON-YYYY'
end-sql
end-procedure
```

DOCUMENT paragraphs

To create form letters, you use a DOCUMENT section. It starts with a BEGIN-DOCUMENT command and ends with an END-DOCUMENT command. In between you lay out the letter and insert variables where you want data from the database to be inserted. SQR will insert the value of the variable when the document is printed. To leave blank lines in a letter, you must explicitly mark them with a .b (see the sample code below).

```
begin-procedure write_letter
begin-document (1,1)
&name
&addr1
&addr2
@city_state_zip
.b
.b
$current-date
Dear Sir or Madam:
.b
Thank you for your recent purchases from ACME Inc. We would like to
tell you about our limited-time offer. During this month, our entire
inventory is marked down by 25%. Yes, you can buy your favorite
merchandise and save too.
To place an order simply dial 800-555-ACME. Delivery is free too, so
don't wait.
.b
.b
Sincerely,
Clark Axelotle
ACME Inc.
end-document
position () @city_state_zip
print &city ()
print ', ' ()
print &state ()
print ' ' ()
print &zip () edit xxxxx-xxxx
new-page
end-procedure ! write_letter
```

PRINT paragraphs

Print paragraphs position the data on the page for printing.

```
begin-procedure print_list           !Name the Procedure
   alter-printer point-size=8 font=4 !change the font
print &project    (+1,1) bold        !Print the line & position
print &street_no  (,5)               !Print the line & position
```

```
print &st_name    (,+1)              !Print the line & position
print &beds       (,30)              !Print the line & position
print &desig      (,37)              !Print the line & position
print &active     (,45)              !Print the line & position
print &sub        (,50)              !Print the line & position
print &vacate_dt  (,55)              !Print the line & position
end-procedure                        !End the Procedure
```

When you select columns from the database in a SELECT paragraph, you can immediately print them using a position. For example:

```
begin-select
phone (,1)
position (+1)
from customers
end-select
```

Family Housing Report
Second Line
Last Line

WV	6140A	NEBRASKA COURT	4	E6 - E9	Y	ADQ	30 JAN 1997
WV	6051B	MINNESOTA COURT	4	E1 - E5	Y	ADQ	16 JUL 1998
WV	6148B	NEBRASKA COURT	4	E6 - E9	Y	ADQ	21 JUN 1999
WV	6149B	OHIO COURT	4	E6 - E9	Y	ADQ	29 JAN 1996
WV	6149A	OHIO COURT	4	E6 - E9	Y	ADQ	12 JUN 2000

Housing Units **4,637**

22-Feb-2001 11:46 AM 88

report_name.SQT

Figure 7 - Resulting report (trimmed for space)

Record Keeping

When creating a report, include an introductory commented paragraph that describes the report and its function along with your name and the date the report was created. When making changes, notate the date and the change that was made to the report.

```
! Title:          Class Report for MCHAS Administrator's Workshop
! Written by:     Author Name
! Written date:   Feb. 21, 2001
! Description:    This is a test report for the documentation for
!                 the MCHAS Administrator Training
!
!Change History: When making changes, enter change info below
! Date:     By:   Change:
! 02-21-01  NNC   Created
```

Creating a Tabular Report

Most SQR reports are cross-tabular reports, which are matrix, or spreadsheet-like, reports. These reports are useful for presenting summary numeric data, as the Waitlist Summary report.

Summary Totals - Waitlists

Waitlist	Waitlist Description	Inactiv	Active	Freeze Zone	Total on List
1BR/E1-E5	1 Bedroom E1 - E5	28	93	9	121
2BR/CG	2 Bedroom Company Grade	0	0	0	0
2BR/E1-E5	2 Bedroom E1 - E5 Midway Park	68	323	32	391
2BR/E1-E5	2 Bedroom E1 - E5 Tarawa Terrace	88	295	29	383
2BR/E6-E9	3 Bedroom E6 - E9	0	1	1	1
3BR/CG-A	3 Bedroom CG Officer Air Station	5	13	3	18
3BR/CG-P	3 Bedroom CG Officer Paradise Point	2	43	4	45
3BR/E1-E5	3 Bedroom E1 - E5	46	135	13	181

Joining Tables

Table joins select related data from two or more tables where they share a common value.

1 to 1 Relationship

A 1 to 1 table join selects data from two tables where only one match will be found. The following example uses the relationship between the OCCUPANT and HOUSING_UNIT tables.

```
begin-procedure get-resident
begin-select
occupant_ssn  &ssn
street_no     &street
  do get-house
from occupant
where assign_dt is not null
end-select
end-procedure

begin-procedure get-house
begin-select
actual_street_no
street_no
street_name
from housing_unit
where street_no = &street
end-select
end-procedure
```

1 to Many Relationship

A 1 to many relationship selects data from two tables where one table may contain multiple entries. The following example uses the relationship between the OCCUPANT and DEPENDENT tables.

```
begin-procedure get-resident
begin-select
occupant_ssn  &ssn
street_no     &street
  do get-family
from occupant
where assign_dt is not null
end-select
end-procedure
```

```
begin-procedure get-family
begin-select
last_name
first_name
relationship
from dependent
where occupant_ssn = &ssn
end-select
end-procedure
```

SQR Functions

For a complete list of SQR commands, refer to the SQR documentation (langref.pdf) in the Docs directory on the SQR CD/ROM. The commands used most often in the body of the report are:

EDIT MASKS - Formats output.

```
     bold
     edit 'DD MON YYYY'
     edit 9,999,999.99
     edit 'xxx-xx-xxxx'
     edit '(xxx)xxx-xxxx'
```

ALTER-PRINTER - Alters printer parameters at run time.

```
ALTER-PRINTER
   [ POINT-SIZE = { point_size_num_lit | _var } ]
   [ FONT-TYPE = { font_type | txt_var } ]
   [ SYMBOL-SET = { symbol_set_id | txt_var } ]
   [ FONT = { font_int_lit | _var } ]
   [ PITCH = { pitch_num_lit | _var } ]
```

CONCAT - Concatenates a variable, column, or literal with a string variable.

```
begin-select
last_name
first_name
middle_init
rank_cd
from occupant
end-select
move rank_cd to $name
concat ‘ ‘ with $name
concat &first_name with $name
concat ‘ ‘ with $name
concat &middle_init with $name
concat ‘ ‘ with $name
concat &last_name with $name

produces SGT JOHN H SMITH
```

You can produce the same results using the **LET** command:

```
let $name = &rank_cd||' '||&first_name||' '||&middle_init||' '||&last_name
```

DO - Invokes the specified procedure.

```
begin-procedure get-resident
begin-select
```

```
occupant_ssn  &ssn
street_no     &street
  do get-house
from occupant
where assign_dt is not null
end-select
end-procedure
```

EVALUATE - Used with WHEN to evaluate a condition.

```
begin-procedure get-beds
begin-select
no_bedrooms  &beds
from housing_unit
end-select
end-procedure

begin-procedure evaluate
     evaluate &beds
      when = 1
        add 1 to #1br
      when = 2
        add 1 to #2br
      when = 3
        add 1 to #3br
      when = 4
      when = 5
        add 1 to #more
     end-evaluate
end-procedure
```

GRAPHIC BOX - graphic (1,1,66) box 58 20
GRAPHIC HORZ-LINE - graphic (4,1,66) horz-line 10
GRAPHIC VERT-LINE - graphic (1,27,54) vert-line

IF - Executes commands depending on the value of a condition.

```
begin-select
phone
  if &phone = ''
    print 'No phone' (,1)
  else
    print &phone (,1)
  end-if
position (+1)
from customers
end-select
```

INPUT - Accepts data entered by the user at a terminal.

```
INPUT $start_date 'Start date for the report (DD MON YYYY)'
INPUT $title 'Enter a Title for your report.'
```

POSITION - Sets the current position on a page.

position (+1) advances the line by one

PRINT - Puts data on the page at a specified position.

```
print 'This is a literal statement'  (1,1) bold
print &phone                         (+1,1) edit '(xxx) xxx-xxxx'
```

WHILE - Begins a WHILE ... END-WHILE loop.

```
while #count < 50
  do get_statistics
    if #stat_count = 100
      break ! Exit WHILE loop.
    end-if
    add 1 to #count
end-while
```

Complex Queries

SQR can process simple or complex queries.

```
Begin-select
o.occupant_ssn
o.last_name
h.street_no
h.street_name
d.last_name
d.first_name
from occupant o, housing_unit h, dependent d
where o.occupant_ssn = d.occupant_ssn
and o.street_no = h.street_no
end-select
```

Notes

Notes

Notes