

NEAR EAST UNIVERSITY



1988

FACULTY OF ENGINEERING

B.Sc. COMPUTER ENGINEERING

GRADUATION PROJECT

TITEL: HOTEL MANAGEMENT

USING FOXPRO PROGRAMMING

NAME: TAISIR MUSTAFA

SUPERVISOR: Mr. OKAN DONANGIL

LEFKOSA-2000

TABLE OF CONTENTS

ACKNOWLEDGMENT.....	1
ABSTRACT.....	2
INTRODUCTION.....	3
PROGRAM EXEPLINATION.....	4
MAIN MENU	19
RESERVATION OFFICE MENU	24
ROOMS SITUATIONS MENU.....	45
• ROOMS.....	49
• RESERVED ROOMS.....	68
• EMPTY ROOMS.....	69
• TEL-LIST SCREEN.....	70
SEARCHING MENU	113
CONCLUSION	144

ACKNOWLEDGMENT

First of all I would like to presents my wishes and thanks to my dears parents whose helped and creates me supportly by standing beside me in all my life to get my goals.

And apical thanks to MY SOPERVISOR: Mr. OKAN DONANGIL.

Also all my thanks to my teachers whose creates me to finish my study as will.

I also would like to thank all those who supported and encouraged me for their help on this project.

The thanks all the thanks to all my friends who helped me to get my goals in my study life, and during this project.

ABSTRACT

Technology In Hotels Management:

The integration of computers and communication technologies allows hotels to control their internal operations (e.g. front office, back office and food and beverage departments) and external operations (e.g. reservation systems, marketing, distribution and links to external data sources) from a single integrated management system. This comprehensive system of information management can become a very powerful tool for wealth creation for the hotel industry. With a comprehensive system in place, all levels of management are supposed to believe to be involved in it and to depend on it to inform most decision-making. There is also integration of the internal computer and communications capabilities of the hotel with external data sources.

The ability of computers to store, process, manipulate and distribute information has greatly improved the efficiency of hotels. By releasing staff time from the mundane paper-pushing functions within the hotel, computers can greatly enhance the opportunities for staff utilization in the hospitality, service and guest-contact points of the hotel. By so doing, the quality of service that hotels offer their guests can be greatly improved.

Image communication is also revolutionizing the way in which hotel bed-nights and other services, such as conference facilities, are sold.

Hotels use computers in their core-information processing centres of marketing and distribution, front office, back-office and food and beverage control.

Information technologies are diffusing in eight key areas of hotel operations:

- Marketing, distribution, reservations and sales;
- Telecommunications;
- Guest accounting;
- Room management;
- Backoffice;
- Food and beverage control; Safety and security.

Introduction

Headlines heralding changes in the hotel business appear in the business and trade press almost daily. "A New Breed of Hotel," says one. "All-Suites Captivate Developers," announces another. "What Is A Resort?" asks a third. A better question might be, "What is the hotel business?" The lodging industry is so broad an endeavor worldwide, divided into so many pieces, that a single definition is not practicable. A single theme is not apparent, and an accurate measurement is not possible.

Nevertheless, many pompously declare it to be the world's seventh- (or second-, etc.) largest industry. Predictions abound that it will soon be the world's largest industry. Undoubtedly, lodging is a major segment of the international economy. As a major component of world tourism, it drives the economic engines of developing countries and accounts for a great deal of investment dollars.

The General Manager:

Management titles vary from hotel to hotel just as hotel organizations do. The chief operating officer, the person responsible for all the departments and for the general profitability of the operation, uses one of several names. Better service and more individualized attention is expected when the manager is also the owner. The term owner-manager or a variation of it, is used to communicate this situation.

A similar association is made if the general manager holds a corporate office. Then the title of vice-president and general manager is used. When the general manager merely sits on the corporate board of directors without holding a corporate position, the title could be director and general manager, or managing director.

Marketing, Distribution and Sales:

Product distribution is a critically important function of hotels. Information technologies, such as computerized reservation systems and video brochures, assist hotels in marketing and distributing their bed-nights. For hotels, their employment of information technologies to link together their front-office and food and beverage departments may be necessary for the efficient and cost-effective delivery of their services. However, it will not be sufficient to guarantee the sale of hotel bed-nights. Without links to international marketing and distribution networks, hotel bed-nights cannot be sold.

If the hotel staff non-users of computer technologies in their internal management functions but, at the same time, to be externally linked to international computerized reservation networks. Similarly, the major technological flagships of the Holiday Corporation are its reservations and communications systems (Holidex 1 Halide 11 and Hi-Net), which are in the distribution, marketing and product-packaging areas of the hotel company's production system. In 1986 Hi-Net was the largest privately owned satellite communications network in the world. Reservations are a key to the sale of hotel bed-nights. Reservation systems, depending on the software and sophistication, contain information and generate reports on:

1. Rooms available by each day;
2. The number of guest expected to arrive and depart each day;
3. Reservations that have changes made to them on a particular day;
4. Cancellations by day;
5. Deposits by guests;
6. Numbers of customers;
7. Date of arrival,

In addition, database can generate forecasts on expected arrivals, departures and rooms sold. Information contained in, and generated by, hotel reservation systems is an invaluable source of marketing information and can generate mailing lists, client profiles and preferences. It can also monitor hotel performance through the development and processing of guest questionnaires. Reservation systems also allow hotels to carry out travel-agency and tour-

operator analyses in order to determine which agencies and operators consistently generate business for the hotels.

Communications:

There are two principal areas in which information technologies are helping hotels to improve their communications:

1. In their internal interdepartmental communications;
2. In their external links with agents, suppliers, reservation systems and data networks.

The internal interdepartmental links of the hotel are facilitated by computers and communications technologies, which serve to integrate the front-office, back-office and food-and-beverage operations. This is facilitated through computer-to-computer communications. The external hotel communication links are necessary between hotel and head office, between hotels and their marketing/distribution agents and between hotel guests and the international environment, for example, entertainment and/or stock-market information.

Communications technologies used by hotels include digital telephone systems, teleconferencing, satellite broadcasting, videotext and audiovisual information tools, image communications and various communication networks for reservations and communications.

Telephone systems used in hotels have been substantially improved to incorporate features such as call accounting systems. These systems can

automatically track calls made by guests so that they can be promptly identified and billed. Least-cost routing options automatically send each call over the cheapest transmission networks. The use of these systems has substantially reduced the need for telephone operators. Automatic call-accounting systems now help hotels to transform their telephone calls into an important profit center of the hotels. As a marketing technique, for example, hotels can offer long-distance telephone calls at discount rates to their guests and still find it profitable.

Teleconferencing involves the transmission of both voice and video via telecommunications technology. It allows conferences and meetings to be held where participants are in different locations. Teleconferencing requires the use of earth stations and satellite respondents, which are both very costly. Two types of teleconferencing facilities are available: Interactive (which allows two-way transmission and receipt of both voice and video and one or more locations) and receive-only (where information flows in only one direction). This technology makes it possible for a hotel conference rooms to become a lecture theatre for delegates on conferences where the main speakers might be elsewhere. It also allows for closer association among hotels situated at different locations.

Satellite communications are also important entertainment tools for the hotels industry. This is especially the case where hotels can increase services to their guests by transmitting live sporting events such as the World Cup, Wimbledon, the Olympic Games and Super Bowl.

In the guest rooms, videotext terminals are connected to the television screen and to the telephone (via a modem) so travelers can gain access to distant information sources. On these terminals, hotel guests can access information on travel and schedules, news reports, updates on stock prices, and many other subjects depending on the networks to which the hotel is connected.

Guest Accounting:

Guest-accounting systems are often contained within the reservation system. They keep track of individual guest charges and can be interfaced with point-of-sale terminals in restaurants, bars and shops throughout the hotel. This greatly increases efficiency and saves potential loss of revenue through unrecorded transactions. The guest history file enables frequent guests to be identified for special attention as it can store data on special requests from previous visits. This could help to encourage repeat business through the provision of special services.

Room Management:

Room-management systems can give updated information on room occupancy and status and they assist in scheduling housekeeper duties for maximum efficiency. Rooming lists, arrivals, stay-overs, extended stays, departures and room preferences can all be handled by room-management systems.

Automatic wake-up calls and message handling are also possible these systems.

Back Office:

The back-office procedures in a hotel are similar to general business applications and include general ledger, payroll, purchasing, inventory control, cash flow and revenue control and management reports. Technology makes the workload easier in the back-office by automating many of the frequent repetitive, data entry and processing functions. And, by extension, while technology saves on labors, it releases valuable human hours for guest contacts.

Food and Beverage:

Food-and-beverage systems allow the hotel to better plan and manage inventory, costing and sales forecasting of its food and beverages. Meals can be better planned and coordinated, and interfaces with guest accounting and back-office systems can increase efficiency. Other technologies available to assist hotels in controlling costs and minimizing theft include electronically controlled drink dispensing machines and robot bars. While point-of-sale interfaces are important in effectively trucking guest purchases at bars and restaurants.

ACCOUNTING:

As previously mentioned, the controller or chief accountant in multiple operations has a dual responsibility- to the general manager, and to the chief financial officer in the home office. The corporate responsibility serves as a definite aid in the performance of a controller's 5 duties. The title suggests one who controls, not merely records, the income and expenses of the hotel. In respect to the income, the position involves checking the accuracy of the amounts reported, thus ensuring that the hotel receives 100 percent of the money spent by the guests.

Checking expenses requires more than the mere verification of the mathematical accuracy of the creditors bills. The legitimacy of the charges, as to both quantity received and price charged, must be checked. Since the general manager or a designated representative should approve all purchases, the controller is in the effect checking the general manager. It is easier, and the results more meaningful, to check the work of a person who does not have the sole authority to hire or fire you.

However, this does not in any way alter the primary function of the controller- to fully cooperate with the general manager and provide all the reports and statistics requested and needed to properly evaluate and operate the property. Most hotels operate on a budget, the preparation of which is the responsibility of the controller. Despite the many elements that are taken into consideration in assembling the figures, a budget is ultimately only an

estimate, an educated guess, of the income and expenses for a given period in the future.

The primary department heads, with the aid of their assistance and sub department heads, are the best qualified to make these forecasts. With these figures as a guide, the controller prepares the budget and presents it to top management, the general manager, and, in multiple operations, the designated corporate officer, for final approval. Once this has been obtained, each member of the management team is obligated to abide by the budget and hold his or her department's expenses to the allocated amount.

RESERVATIONS:

SYSTEMS AND OPERATIONS:

Where the owners and management of the participating hotels are varied and wide spread, it naturally follows that the operating procedures used in the hotels are also varied. A reservation system operates on the premise that every hotel in the system will communicate with the use of system in the same fashion.

Charges to an individual hotel using a system are for the most part generated by the reservations taken for that specific property. Most hotels are reluctant to pay for a reservation the guest does not show up, so in many cases billings to the participating hotels permitted the deduction of charges for "no show". The validity of such deduction can be verified only by examination of the hotel's records or direct communication with the source of reservation.

HOUSE KEEPING:

The dictionary definition of a housekeeper is "one who does or oversees the work of keeping house." This explanation states the principle function of all housekeepers. In private family homes, "keeping house" includes taking care of the kitchen. In a hotel this responsibility rarely includes the kitchen, pantries, refrigerators, storage rooms for groceries, kitchen-cleaning and related supplies, and the dish-and ware-washing areas. The maintenance, cleaning, and daily operations of these are under the supervision of the chef or chief steward.

The rest of hotel, all under the direct supervision of the executive housekeeper, can be divided into eight general areas:

- guest rooms.
- Halls and corridors.
- Lobby.
- Public rooms and restaurants.
- Offices.
- Stairways.
- Windows.
- Stores, concessions, and other leased areas.

RECEPTION DETAILES:

REGISTRATION DOCUMENT:

- Name and Surname of Customer
- Room Number
- Address of customer
- Arrive and Departure dates
- NAME OF THE CUSTOMER.
- Type of Service
 - Notices
 - Writing of "The rates are includes the taxes"
 - Writing of "The Management of the hotel will not be responsibility for the any losing things (like money, jewellery, gold)"
 - On the top of the right corner; "Logo and name of hotel"

RESERVATIONS AND DETAILS:

RESERVATION DOCUMENT

- Name and Surname of the Reservationer
- Number of the person
- Arrive and Departure Dates
- Address of the Reservationer

- Name of the Travel Agency
- Date of the Reservation

DOCUMENTS OF ROOMS

CLASSIFICATION OF ROOMS

- According to the Quality
 - First Quality { Kingdome Room and just one room)
 - Second Quality {Suit Rooms up most tree room)
 - Thirty Quality { Normal Rooms)
- According to the Number Person For 1 person
 - For 2 person For 3 person
 - For 4 person

After we classified our rooms as below with their wife, and we should see that which rooms are empty or not, which rooms are reserved, within the which dates and who reserved.

And I could do that which is the in the different way of the searching like according to the numbers of the persons or according to the name or surname of customer, or the date of arrival or departure or according to the rooms are empty or tull.

FORMULATING THE TYPES OF SERVICES

- Bed end Breakfast:

- The Price of the Breakfast times each day
- Half Port:
Bedded Breakfast + Price of dinner times each day.
- Full Port:
Half part + Price of noon eating.

CUSTOMER EXTRAS

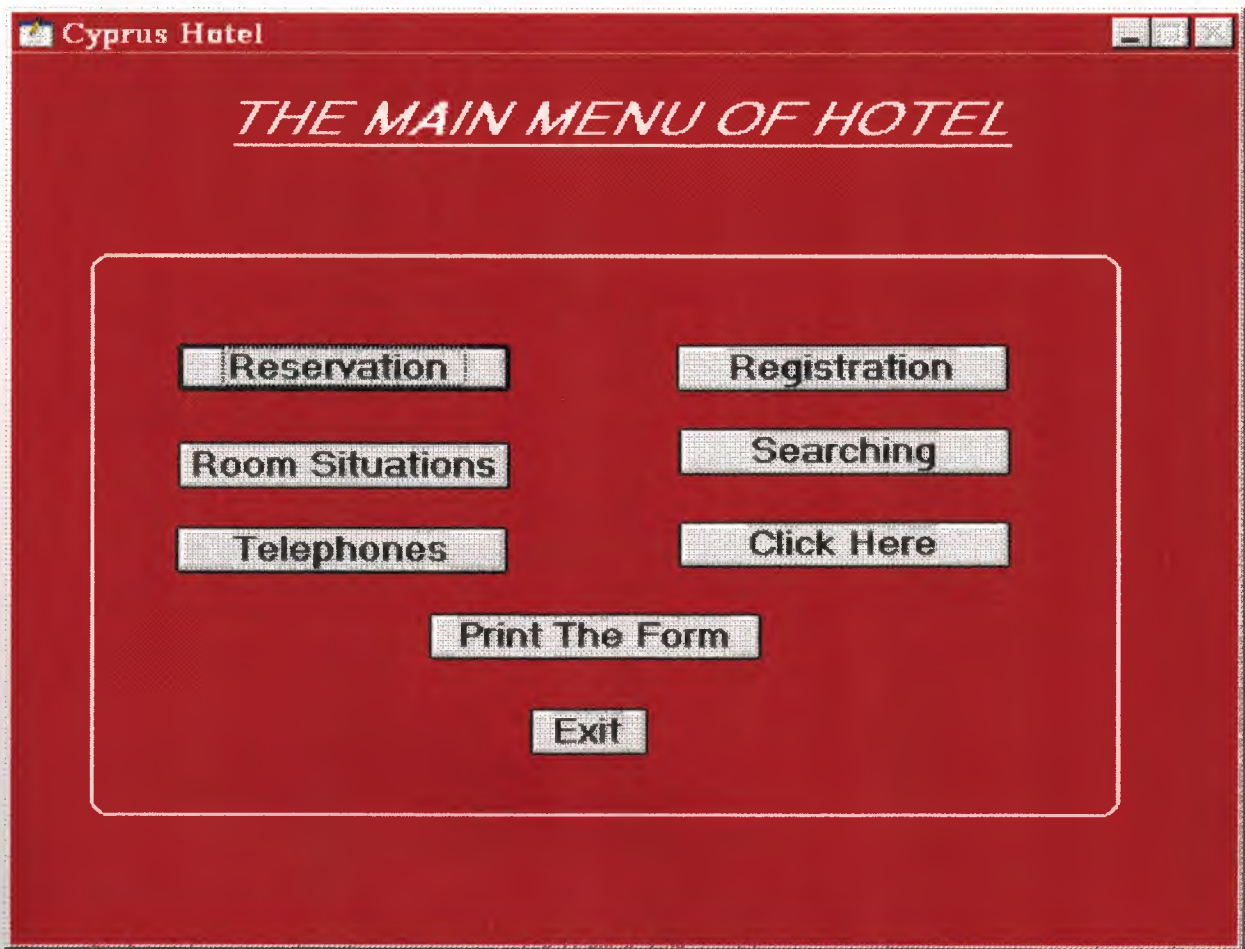
CUSTOMER EXTRA DOCUMENT

- Name of Department
- Name and Surname of Cashier and Signature
- Name and Surname of Manager and Signature
- Name and Surname of Customer and
- Room Number
- Date and Time of extra
- Detail of account Bill of account

After we create classers that consist of from this type of documentation we can divide it to the files. And if we entered the name of customer's or the room number to check the room

account whether it has the extras or not it should be able to give us immediately.

MAIN MENU:



```

SET TALK OFF
IF NOT WEXIST("_02o19fjdo")
    DEFINE WINDOW _02o19fjdo ;
        AT 0.000, 0.000 ;
        SIZE 33.846,106.200 ;
        TITLE "Cyprus Hotel" ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        NOCLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,,128,0,0)
    MOVE WINDOW _02o19fjdo CENTER
ENDIF

```

```

#REGION 1
IF WVISIBLE("_02o19fjdo")
    ACTIVATE WINDOW _02o19fjdo SAME
ELSE
    ACTIVATE WINDOW _02o19fjdo NOSHOW
ENDIF
@ 1.308,17.600 SAY "THE MAIN MENU OF HOTEL" ;
    FONT "MS Sans Serif", 18 ;
    STYLE "BIT" ;
    COLOR RGB(255,255,255,,,,)
@ 11.077,14.400 GET a ;
    PICTURE "@*HN Reservation " ;
    SIZE 1.150,16.000,0.444 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 12 ;
    STYLE "B" ;
    VALID bt('Reservation')
@ 11.077,57.600 GET a ;
    PICTURE "@*HN Registration" ;
    SIZE 1.150,16.000,0.444 ;

```

DEFAULT 1 ;
 FONT "MS Sans Serif", 12 ;
 STYLE "B" ;
 VALID bt('Registration')

@ 14.231,57.800 GET a ;
 PICTURE "@*HN Searching" ;
 SIZE 1.150,16.000,0.444 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 12 ;
 STYLE "B" ;
 VALID bt('Searching')

@ 18.000,14.200 GET a ;
 PICTURE "@*HN Telephones" ;
 SIZE 1.150,16.000,0.444 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 12 ;
 STYLE "B" ;
 VALID bt('Tel')

@ 14.769,14.400 GET a ;
 PICTURE "@*HN Room Situations" ;
 SIZE 1.150,16.000,0.444 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 12 ;
 STYLE "B" ;
 VALID bt('Room')

@ 21.308,36.200 GET a ;
 PICTURE "@*HN Print The Form" ;
 SIZE 1.150,16.000,0.444 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 12 ;
 STYLE "B" ;
 VALID bt('Print')

@ 24.923,45.000 GET m.bt_exit ;
 PICTURE "@*HN Exit" ;
 SIZE 1.150,5.667,0.444 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 12 ;
 STYLE "B" ;
 VALID bt('EXIT')

@ 17.769,57.800 GET a ;

```

    PICTURE "@*HN Click Here" ;
    SIZE 1.150,16.000,0.444 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 12 ;
    STYLE "B" ;
    VALID bt('Click')
@ 3.385,19.200 TO 3.385,86.800 ;
    PEN 2, 8 ;
    STYLE "1" ;
    COLOR RGB(255,255,255,255,255,255)
@ 7.615,6.800 TO 29.077,96.200 ;
    PEN 2, 8 ;
    STYLE "16" ;
    COLOR RGB(255,255,255,,,)

```

```

IF NOT WVISIBLE("_02o19fjdo")
    ACTIVATE WINDOW _02o19fjdo
ENDIF

```

READ CYCLE

```


PROCEDURE BT
    PARAMETER m.btnname
    DO CASE
CASE m.btnname='Reservation'
    DO c:\taisir\reserv.prg
CASE m.btnname='Registration'
    DO c:\taisir\register.spr
CASE m.btnname='Tel'
    DO c:\taisir\phone.prg
CASE m.btnname='Searching'
    DO c:\taisir\search.prg
CASE m.btnname='Room'
    DO c:\taisir\room.prg
CASE m.btnname='Print'
    DO C:\taisir\form.prg
CASE m.btnname='Click'
    DO c:\taisir\cover.prg
CASE m.btnname='EXIT'
    set sysm to defa

```


DEACTIVATE WINDOW _02o19fjdo
RELEASE WINDOW _02o19fjdo
CLOSE ALL WINDOWS
SET CLOCK OFF

ENDCASE
RETURN

1- RESERVATION OFFICE MENU:

 Cyprus Hotel

RESERVATION OFFICE

Room No:

1

Room Type:

Single

Room Price:

20

\$

Name:

sam

Surname:

rody

Sex:

☒ Male

☐ Female

Telephone:

2235677

Nationality:

british

First Date:

04/23/99

Last Date:

05/01/99

Top

Prev

Next

End

Locate

Add

Edit

Delete

Print

Close

```

* *****
*
* 05/04/00          RESERV.PR          21:24:59
*
* *****
*
* Author's Name
*
* Copyright (c) 2000 Company Name
* Address
* City, Zip
*
* Description:
* This program was automatically generated by GENSCRN.
*
* *****

```

```

* *****
*
* RESERV/Windows Setup Code - SECTION 1
*
* *****
*

```

```

#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF
m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"

```

```
SET TALK ON
ENDIF
```

```
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
```

```
IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
```

```
m.rborder = SET("READBORDER")
SET READBORDER ON
```

```
m.currarea = SELECT()
```

```
* *****
* *
* *   S7099505/Windows Databases, Indexes, Relations
* *
* *****
```

```
IF USED("reserv")
    SELECT reserv
    SET ORDER TO TAG "_sap0k3oyy"
ELSE
    SELECT 0
    USE (LOCFILE("C:\taisir\reserv.dbf", "DBF", "Where is reserv?"));
    AGAIN ALIAS reserv ;
    ORDER TAG "_sap0k3oyy"
ENDIF
```

```

* *****
*      *      Windows Window definitions
*      *
* *****

```

```

IF NOT WEXIST("_02o19wid1")
    DEFINE WINDOW _02o19wid1 ;
        AT 0.000, 0.000 ;
        SIZE 26.077,83.333 ;
        TITLE "Cyprus Hotel" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        COLOR RGB(,,128,0,0)
    MOVE WINDOW _02o19wid1 CENTER
ENDIF

```

```

* *****
*      *
*      *      RESERV/Windows Setup Code - SECTION 2
*      *
* *****

```

```

#REGION 1
#DEFINE C_DBFEMPTY          'Database is empty, add a record?'
#DEFINE C_EDITS              'Please finish your edits.'
#DEFINE C_TOPFILE            'Top of file.'
#DEFINE C_ENDFILE            'End of file.'
#DEFINE C_BRTITLE            'Locate Record'
#DEFINE C_NOLOCK              'Sorry, could not lock record -- try again later.'
#DEFINE C_ECANCEL            'Edits Canceled.'
#DEFINE C_DELREC              'Delete selected record?'
#DEFINE C_NOFEAT              'Feature not available yet.'
#DEFINE C_NOWIZ               'Wizard application is not available.'
#DEFINE C_MAKEREPO            'Creating report with Report Wizard.'
#DEFINE C_NOREPO              'Could not create report.'
#DEFINE C_DELNOTE             'Deleting records...'
#DEFINE C_READONLY           'Table is read-only. No editing allowed.'

```



```

#DEFINE C_NOTABLE      'No table selected. Open table or run query.'
#DEFINE C_BADEXPRESS   'Invalid expression.'
#DEFINE C_LOCWIZ        'Locate WIZARD.APP:'
#DEFINE C_MULTITABLE    'You have multiple related tables. Adding
records in not allowed.'
MOVE WINDOW '_02o19wid1' CENTER
PRIVATE isediting,isadding,wztblarr
PRIVATE wzoldddelete,wzolderror,wzoldesc
PRIVATE wzalias, tempcurs,wzlastrec
PRIVATE isreadonly,find_drop,is2table
IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF
m.wztblarr= "
m.wzalias=SELECT()
m.isediting=.F.
m.isadding=.F.
m.is2table = .F.
m.wzoldddelete=SET('DELETE')
SET DELETED ON
m.tempcurs=SYS(2015) &&used if General field
m.wzlastrec = 1
m.wzolderror=ON('error')
ON ERROR DO wizerrorhandler
wzoldesc=ON('KEY','ESCAPE')
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)
m.isreadonly=IIF(ISREAD(),.T.,.F.)
IF m.isreadonly
    WAIT WINDOW C_READONLY TIMEOUT 1
ENDIF

IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)
    APPEND BLANK
ENDIF

GOTO TOP
SCATTER MEMVAR MEMO

```

```

* *****
*
*      RESERV/Windows Screen Layout
*
* *****

```

```
#REGION 1
```

```

IF WVISIBLE("_02o19wid1")
    ACTIVATE WINDOW _02o19wid1 SAME
ELSE
    ACTIVATE WINDOW _02o19wid1 NOSHOW
ENDIF

```

```

@ 0.538,22.500 SAY "RESERVATION OFFICE" ;
    FONT "MS Sans Serif", 14 ;
    STYLE "BT" ;
    COLOR RGB(255,255,255,,,)

```

```

@ 2.692,0.000 TO 2.692,83.333 ;
    PEN 2, 8 ;
    STYLE "1" ;
    COLOR RGB(255,255,255,255,255,255)

```

```

@ 22.231,0.000 TO 22.231,83.333 ;
    PEN 2, 8 ;
    STYLE "1" ;
    COLOR RGB(255,255,255,255,255,255)

```

```

@ 3.692,24.667 SAY "Room No:" ;
    SIZE 0.938,14.200 ;
    FONT "Times New Roman", 9 ;
    STYLE "BT" ;
    PICTURE "@J" ;
    COLOR RGB(,,255,255,255)

```

```

@ 3.769,39.167 GET m.room_no ;
    SIZE 1.000,12.000 ;
    DEFAULT " " ;
    FONT "Times New Roman", 8 ;
    STYLE "B" ;
    PICTURE "@K 99999" ;
    WHEN isediting ;
    COLOR ,RGB(0,0,0,255,255,255)

```

```

@ 5.538,24.667 SAY "Room Type:" ;
    SIZE 0.938,14.200 ;

```

```

FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,,255,255,255)
@ 5.615,39.167 GET m.room_type ;
SIZE 1.000,16.000 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K XXXXXXXXXXXX" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 7.385,24.667 SAY "Room Price:" ;
SIZE 0.938,14.200 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,,255,255,255)
@ 7.462,39.167 GET m.room_price ;
SIZE 1.000,15.600 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K 9,999,999,999" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 7.462,54.833 GET m.MONEY ;
SIZE 1.000,3.000 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K XX" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 9.231,24.667 SAY "Name:" ;
SIZE 0.938,14.200 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,,255,255,255)

```

@ 9.308,39.167 GET m.name ;
 SIZE 1.000,33.000 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 11.077,24.667 SAY "Surname:" ;
 SIZE 0.938,14.200 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,,255,255,255)
 @ 11.154,39.167 GET m.surname ;
 SIZE 1.000,33.000 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 12.923,24.667 SAY "Sex:" ;
 SIZE 0.938,14.200 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,,255,255,255)
 @ 13.000,39.000 GET m.sex ;
 PICTURE "@*RHN Male;Female" ;
 SIZE 1.308,11.167,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "BT"
 @ 14.769,24.667 SAY "Telephone:" ;
 SIZE 0.938,14.200 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,,255,255,255)

@ 14.846,39.167 GET m.telephone ;
 SIZE 1.000,15.600 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 9999999999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 16.615,24.667 SAY "Nationality:" ;
 SIZE 0.938,14.200 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,255,255,255)
 @ 16.692,39.167 GET m.nation ;
 SIZE 1.000,19.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K XXXXXXXXXXXXXXX" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 18.462,24.667 SAY "First Date:" ;
 SIZE 0.938,14.200 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,255,255,255)
 @ 18.538,39.167 GET m.first_date ;
 SIZE 1.000,9.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 20.308,24.667 SAY "Last Date:" ;
 SIZE 0.938,14.200 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;


```

    PICTURE "@J" ;
    COLOR RGB(,,255,255,255)
@ 20.385,39.167 GET m.last_date ;
    SIZE 1.000,9.200 ;
    DEFAULT " " ;
    FONT "Times New Roman", 8 ;
    STYLE "B" ;
    PICTURE "@K" ;
    WHEN isediting ;
    COLOR ,RGB(0,0,0,255,255,255)
@ 23.385,1.667 GET m.top_btn ;
    PICTURE "@*HN \<Top" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('TOP') ;
    MESSAGE 'Go to first record.'
@ 23.385,9.667 GET m.prev_btn ;
    PICTURE "@*HN \<Prev" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('PREV') ;
    MESSAGE 'Go to previous record.'
@ 23.385,17.667 GET m.next_btn ;
    PICTURE "@*HN \<Next" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('NEXT') ;
    MESSAGE 'Go to next record.'
@ 23.385,25.667 GET m.end_btn ;
    PICTURE "@*HN \<End" ;
    SIZE 1.769,7.833,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;

```

```

VALID btn_val('END') ;
MESSAGE 'Go to last record.'
@ 23.385,33.667 GET m.loc_btn ;
PICTURE "@*HN \<Locate" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('LOCATE') ;
MESSAGE 'Locate a record.'
@ 23.385,41.667 GET m.add_btn ;
PICTURE "@*HN \<Add" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('ADD') ;
MESSAGE 'Add a new record.'
@ 23.385,49.667 GET m.edit_btn ;
PICTURE "@*HN Ed\<it" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('EDIT') ;
MESSAGE 'Edit current record.'
@ 23.385,57.667 GET m.del_btn ;
PICTURE "@*HN \<Delete" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('DELETE') ;
MESSAGE 'Delete current record.'
@ 23.385,65.667 GET m.prnt_btn ;
PICTURE "@*HN P\<rint" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;

```

```

        VALID btn_val('PRINT') ;
        MESSAGE 'Print report.'
@ 23.385,73.667 GET m.exit_btn ;
        PICTURE "@*HN \<Close" ;
        SIZE 1.769,7.833,0.667 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        VALID btn_val('EXIT') ;
        MESSAGE 'Close screen.'

```

```

IF NOT WVISIBLE("_02o19wid1")
        ACTIVATE WINDOW _02o19wid1
ENDIF

```

```

*      *****
*      *
*      *   WindowsREAD contains clauses from SCREEN s7099505
*      *
*      *****
*

```

```

READ CYCLE ;
        ACTIVATE READACT() ;
        DEACTIVATE READDEAC() ;
        NOLOCK

```

```

RELEASE WINDOW _02o19wid1

```

```

*      *****
*      *
*      *   Windows Closing Databases
*      *
*      *****
*

```

```

IF USED("reserv")
        SELECT reserv
        USE
ENDIF

```

```
SELECT (m.currarea)
#REGION 0
```

```
SET READBORDER &rborder
```

```
IF m.talkstat = "ON"
    SET TALK ON
```

```
ENDIF
```

```
IF m.compstat = "ON"
    SET COMPATIBLE ON
```

```
ENDIF
```

```
* *****
*
*      RESERV/Windows Cleanup Code
*
* *****
*
```

```
#REGION 1
```

```
SET DELETED &wzolddelete
```

```
SET FIELDS &wzfields
```

```
ON ERROR &wzolderror
```

```
ON KEY LABEL ESCAPE &wzoldesc
```

```
DO CASE
```

```
CASE _DOS AND SET('DISPLAY')='VGA25'
```

```
    @24,0 CLEAR TO 24,79
```

```
CASE _DOS AND SET('DISPLAY')='VGA50'
```

```
    @49,0 CLEAR TO 49,79
```

```
CASE _DOS
```

```
    @24,0 CLEAR TO 24,79
```

```
ENDCASE
```

```
****Procedures****
```

```
* *****
*
*      RESERV/Windows Supporting Procedures and Functions
*
* *****
```


*

#REGION 1

PROCEDURE readdeac

IF isediting

ACTIVATE WINDOW '_02o19wid1'

WAIT WINDOW C_EDITS NOWAIT

ENDIF

IF !WVISIBLE(WOUTPUT())

CLEAR READ

RETURN .T.

ENDIF

RETURN .F.

PROCEDURE readact

IF !isediting

SELECT (m.wzalias)

SHOW GETS

ENDIF

DO REFRESH

RETURN

PROCEDURE wizerrorhandler

* This very simple error handler is primarily intended

* to trap for General field OLE errors which may occur

* during editing from the MODIFY GENERAL window.

WAIT WINDOW message()

RETURN

PROCEDURE printrec

PRIVATE sOldError,wizfname,saverec,savearea,tmpcurs,tmpstr

PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record

STORE 1 TO p_recs,p_output

STORE 0 TO prnt_btn

STORE RECNO() TO saverec

m.sOldError=ON('error')

DO pdialog

IF m.prnt_btn = 2

RETURN

ENDIF

IF !FILE(ALIAS()+'FRX')

m.wizfname=SYS(2004)+'WIZARDS\'+ 'WIZARD.APP'

IF !FILE(m.wizfname)


```

        ON ERROR *
        m.wizfname=LOCFILE('WIZARD.APP','APP',C_LOCWIZ)
        ON ERROR &sOldError
        IF !'WIZARD.APP'$UPPER(m.wizfname)
        WAIT WINDOW C_NOWIZ
        RETURN
        ENDIF
    ENDIF
WAIT WINDOW C_MAKEREPO NOWAIT
    m.savearea=SELECT()
    m.tmpcurs='_'+LEFT(SYS(3),7)
    CREATE CURSOR (m.tmpcurs) (comment m)
    m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)
    INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)
    SELECT (m.savearea)
    DO (m.wizfname) WITH
    ", 'WZ_QREPO', 'NOSCRN/CREATE', ALIAS(), m.tmpcurs
    USE IN (m.tmpcurs)
WAIT CLEAR
    IF !FILE(ALIAS()+'.FRX') && wizard could not create report
    WAIT WINDOW C_NOREPO
    RETURN
    ENDIF
ENDIF
m.pr_out=IIF(m.p_output=1, 'TO PRINT NOCONSOLE', 'PREVIEW')
m.pr_record=IIF(m.p_recs=1, 'NEXT 1', 'ALL')
REPORT FORM (ALIAS()) &pr_out &pr_record
GO m.saverec
RETURN
PROCEDURE BTN_VAL
    PARAMETER m.btnname
    DO CASE
    CASE m.btnname='TOP'
        GO TOP
        WAIT WINDOW C_TOPFILE NOWAIT
    CASE m.btnname='PREV'
        IF !BOF()
            SKIP -1
        ENDIF
        IF BOF()

```

```

        WAIT WINDOW C_TOPFILE NOWAIT
        GO TOP
    ENDIF
CASE m.btnname='NEXT'
    IF !EOF()
        SKIP 1
    ENDIF
    IF EOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        GO BOTTOM
    ENDIF
CASE m.btnname='END'
    GO BOTTOM
    WAIT WINDOW C_ENDFILE NOWAIT
CASE m.btnname='LOCATE'
    DO loc_dlog
CASE m.btnname='ADD' AND !isediting &&add record
    isediting=.T.
    isadding=.T.
    =edithand('ADD')
    _curobj=1
    DO refresh
    SHOW GETS
    RETURN
CASE m.btnname='EDIT' AND !isediting &&edit record
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF RLOCK()
        isediting=.T.
        _curobj=1
        DO refresh
        RETURN
    ELSE
        WAIT WINDOW C_NOLOCK
    ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
    IF isadding
        =edithand('SAVE')

```

```

ELSE
    GATHER MEMVAR MEMO
ENDIF
UNLOCK
isediting=.F.
isadding=.F.
DO refresh
CASE m.btnname='DELETE' AND isediting &&cancel record
    IF isadding
        =edithand('CANCEL')
    ENDIF
    isediting=.F.
    isadding=.F.
    UNLOCK
    WAIT WINDOW C_ECANCEL NOWAIT
    DO refresh
CASE m.btnname='DELETE'
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF fox_alert(C_DELREC)
        DELETE
        IF !EOF() AND DELETED()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C_ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    ENDIF
CASE m.btnname='PRINT'
    DO printrec
    RETURN
CASE m.btnname='EXIT'
    m.bailout=.T.      &&this is needed if used with FoxApp
    CLEAR READ
    RETURN
ENDCASE
SCATTER MEMVAR MEMO

```

```

        SHOW GETS
RETURN
PROCEDURE REFRESH
DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting
    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE
    SHOW GET loc_btn DISABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE
    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "\<Save"
    SHOW GET del_btn,1 PROMPT "\<Cancel"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
    RETURN
OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"
    SHOW GET del_btn,1 PROMPT "\<Delete"
    SHOW GETS ENABLE
ENDCASE
IF m.is2table
    SHOW GET add_btn DISABLE
ENDIF
ON KEY LABEL ESCAPE
RETURN
PROCEDURE edithand

```



```

PARAMETER m.paction
* procedure handles edits
DO CASE
CASE m.paction = 'ADD'
    SCATTER MEMVAR MEMO BLANK
CASE m.paction = 'SAVE'
    INSERT INTO (ALIAS()) FROM MEMVAR
CASE m.paction = 'CANCEL'
    * nothing here
ENDCASE
RETURN

PROCEDURE fox_alert
PARAMETER wzalrtmess
PRIVATE alrtbtn
m.alrtbtn=2
    DEFINE WINDOW _qec1ij2t7 AT 0,0 SIZE 8,50 ;
    FONT "MS Sans Serif",10 STYLE 'B' ;
    FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()
    MOVE WINDOW _qec1ij2t7 CENTER
    ACTIVATE WINDOW _qec1ij2t7 NOSHOW
    @ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;
    FONT "MS Sans Serif", 10 STYLE "B"
    @ 6,18 GET m.alrtbtn ;
    PICTURE "@*HT \<OK;\?!\<Cancel" ;
    SIZE 1.769,8.667,1.333 ;
    FONT "MS Sans Serif", 8 STYLE "B"
    ACTIVATE WINDOW _qec1ij2t7
    READ CYCLE MODAL
    RELEASE WINDOW _qec1ij2t7
RETURN m.alrtbtn=1
PROCEDURE pdialog
    DEFINE WINDOW _qjn12zbvh ;
    AT 0.000, 0.000 ;
    SIZE 13.231,54.800 ;
    TITLE "Microsoft FoxPro" ;
    FONT "MS Sans Serif", 8 ;
    FLOAT NOCLOSE MINIMIZE SYSTEM
    MOVE WINDOW _qjn12zbvh CENTER
    ACTIVATE WINDOW _qjn12zbvh NOSHOW

```



```

@ 2.846,33.600 SAY "Output:" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 2.846,4.800 SAY "Print:" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 4.692,7.200 GET m.p_recs ;
    PICTURE "@*RVN \<Current Record;\<All Records" ;
    SIZE 1.308,18.500,0.308 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 4.692,36.000 GET m.p_output ;
    PICTURE "@*RVN \<Printer;Pre\<view" ;
    SIZE 1.308,12.000,0.308 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 10.154,16.600 GET m.prnt_btn ;
    PICTURE "@*HT P\<rint;Ca\<ncel" ;
    SIZE 1.769,8.667,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B"
    ACTIVATE WINDOW _qjn12zbvh
    READ CYCLE MODAL
    RELEASE WINDOW _qjn12zbvh
RETURN

PROCEDURE loc_dlog
    PRIVATE gfields,i
    DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;
        SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans
Serif",8
    MOVE WINDOW wzlocate CENTER
    m.gfields=SET('FIELDS',2)
    IF !EMPTY(RELATION(1))
        SET FIELDS ON
        IF m.gfields # 'GLOBAL'

```

```

        SET FIELDS GLOBAL
    ENDIF
    IF EMPTY(FLDLIST())
        m.i=1
        DO WHILE !EMPTY(OBJVAR(m.i))
            IF ATC('M.',OBJVAR(m.i))=0
                SET FIELDS TO (OBJVAR(m.i))
            ENDIF
            m.i = m.i + 1
        ENDDO
    ENDIF
ENDIF
ENDIF
BROWSE WINDOW wzlocate NOEDIT NODELETE ;
    NOMENU TITLE C _BRTITLE
SET FIELDS &gfields
SET FIELDS OFF
RELEASE WINDOW wzlocate
RETURN

```

3- ROOMS SITUATIONS MENU:



```

SET TALK OFF
IF NOT WEXIST("_02o1b7nbd")
    DEFINE WINDOW _02o1b7nbd ;
        AT 0.000, 0.000 ;
        SIZE 22.308,56.000 ;
        TITLE "Cyprus Hotel" ;
        FONT "MS Sans Serif", 8 ;
        FLOAT ;
        NOCLOSE ;
        MINIMIZE ;
        SYSTEM ;
        COLOR RGB(,,128,0,0)
    MOVE WINDOW _02o1b7nbd CENTER
ENDIF

#REGION 1
IF WVISIBLE("_02o1b7nbd")
    ACTIVATE WINDOW _02o1b7nbd SAME
ELSE
    ACTIVATE WINDOW _02o1b7nbd NOSHOW
ENDIF

@ 0.769,7.400 SAY "Rooms Situations" ;
    FONT "MS Sans Serif", 18 ;
    STYLE "BT" ;
    COLOR RGB(255,255,255,,,)
@ 2.769,7.200 TO 2.769,48.600 ;
    PEN 2, 8 ;
    STYLE "I" ;
    COLOR RGB(255,255,255,255,255,255)
@ 8.692,12.200 GET a ;
    PICTURE "@*HN Rooms" ;
    SIZE 1.150,16.111,0.444 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 12 ;
    STYLE "B" ;
    VALID bt('Room')
@ 11.462,12.200 GET a ;
    PICTURE "@*HN Reserved Rooms" ;
    SIZE 1.150,16.111,0.444 ;
    DEFAULT 1 ;

```

```

    FONT "MS Sans Serif", 12 ;
    STYLE "B" ;
    VALID bt('Reserved')
@ 14.231,12.200 GET a ;
    PICTURE "@*HN Empty Rooms" ;
    SIZE 1.150,16.111,0.444 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 12 ;
    STYLE "B" ;
    VALID bt('Empty')
@ 17.000,19.200 GET a ;
    PICTURE "@*HN Close" ;
    SIZE 1.150,6.778,0.444 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 12 ;
    STYLE "B" ;
    VALID bt('EXIT')
@ 6.538,4.200 TO 20.615,50.200 ;
    PEN 2, 8 ;
    STYLE "16" ;
    COLOR RGB(255,255,255,,,)

```

```

IF NOT WVISIBLE("_02o1b7nbd")
    ACTIVATE WINDOW _02o1b7nbd
ENDIF

```

READ CYCLE

```


PROCEDURE BT
    PARAMETER m.btnname
    DO CASE
CASE m.btnname='Room'
    DO c:\taisir\rooms.prg
CASE m.btnname='Reserved'
    DO c:\taisir\Res.qpr
CASE m.btnname='Empty'
    DO c:\taisir\emp.qpr
CASE m.btnname='EXIT'
    set sysm to defa
DEACTIVATE WINDOW _

```


RELEASE WINDOW _02o1b7nbd
CLOSE ALL WINDOWS _02o1b7nbd
SET CLOCK OFF

ENDCASE
RETURN

4- ROOM SITUATION SCREEN:

 Cyprus Hotel

ROOM SITUATION

Room No:

Phone No:

Room Price: \$

Situation: ☐ Empty
☐ Full
☒ Reserved

Room Type: ☒ Single
☐ Double
☐ Family

Top

Prev

Next

End

Locate

Add

Edit

Delete

Print

Close

```
PRIVATE wzolddelete,wzolderror,wzoldesc
PRIVATE wzalias, tempcurs,wzlastrec
PRIVATE isreadonly,find_drop,is2table
```

```
IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF
```

```
m.wztblarr= "
m.wzalias=SELECT()
m.isediting=.F.
m.isadding=.F.
m.is2table = .F.
m.wzolddelete=SET('DELETE')
SET DELETED ON
m.tempcurs=SYS(2015) &&used if General field
m.wzlastrec = 1
m.wzolderror=ON('error')
ON ERROR DO wizerrorhandler
wzoldesc=ON('KEY','ESCAPE')
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)
```

```
m.isreadonly=IIF(ISREAD(),.T.,.F.)
IF m.isreadonly
    WAIT WINDOW C_READONLY TIMEOUT 1
ENDIF
```

```
IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)
    APPEND BLANK
ENDIF
```

```
GOTO TOP
SCATTER MEMVAR MEMO
```

```
* *****
* *
* *      ROOMS/Windows Screen Layout
```

```
*
*
* *****
*
```

```
#REGION 1
IF WVISIBLE("_sat0xy4uf")
    ACTIVATE WINDOW _sat0xy4uf SAME
ELSE
    ACTIVATE WINDOW _sat0xy4uf NOSHOW
ENDIF
@ 0.538,26.833 SAY "ROOM SITUATION " ;
    FONT "Times New Roman", 14 ;
    STYLE "BT";
    COLOR RGB(255,255,255,,,)
@ 2.692,0.000 TO 2.692,83.333 ;
    PEN 2, 8 ;
    STYLE "1";
    COLOR RGB(255,255,255,,,)
@ 16.385,0.000 TO 16.385,83.333 ;
    PEN 2, 8 ;
    STYLE "1";
    COLOR RGB(255,255,255,,,)
@ 4.385,12.167 SAY "Room No:" ;
    SIZE 0.938,13.800 ;
    FONT "Times New Roman", 9 ;
    STYLE "BT" ;
    PICTURE "@J" ;
    COLOR RGB(,,255,255,255)
@ 4.385,25.500 GET m.room_no ;
    SIZE 1.000,7.400 ;
    DEFAULT " " ;
    FONT "Times New Roman", 8 ;
    STYLE "B" ;
    PICTURE "@K 999" ;
    WHEN isediting ;
    COLOR ,RGB(0,0,0,255,255,255)
@ 4.385,44.333 SAY "Situation:" ;
    SIZE 0.938,14.200 ;
    FONT "Times New Roman", 9 ;
    STYLE "BT" ;
```



PICTURE "@J" ;
COLOR RGB(,,255,255,255)
@ 4.308,58.833 GET m.situation ;
PICTURE "@*RVN Empty;Full;Reserved" ;
SIZE 1.308,14.667,0.308 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "BT" ;
WHEN isediting
@ 7.231,11.500 SAY "Phone No:" ;
SIZE 0.938,14.200 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,255,255,255)
@ 7.231,25.500 GET m.phone_no ;
SIZE 1.000,6.000 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K 9999" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 10.231,11.500 SAY "Room Price:" ;
SIZE 0.938,14.200 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,255,255,255)
@ 10.308,25.500 GET m.room_price ;
SIZE 1.000,12.000 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K 99999999" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 10.308,37.500 GET m.money ;
SIZE 1.000,3.000 ;
DEFAULT " " ;


```

FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K XX" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 10.308,44.333 SAY "Room Type:" ;
SIZE 0.938,14.200 ;
FONT "Times New Roman", 9 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(,,,255,255,255)
@ 10.231,58.833 GET m.room_type ;
PICTURE "@*RVN Single;Double;Family" ;
SIZE 1.308,11.167,0.308 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "BT" ;
WHEN isediting
@ 17.538,1.667 GET m.top_btn ;
PICTURE "@*HN \<Top" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('TOP') ;
MESSAGE 'Go to first record.'
@ 17.538,9.667 GET m.prev_btn ;
PICTURE "@*HN \<Prev" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('PREV') ;
MESSAGE 'Go to previous record.'
@ 17.538,17.667 GET m.next_btn ;
PICTURE "@*HN \<Next" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;

```

```

VALID btn_val('NEXT') ;
MESSAGE 'Go to next record.'
@ 17.538,25.667 GET m.end_btn ;
PICTURE "@*HN \<End" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('END') ;
MESSAGE 'Go to last record.'
@ 17.538,33.667 GET m.loc_btn ;
PICTURE "@*HN \<Locate" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('LOCATE') ;
MESSAGE 'Locate a record.'
@ 17.538,41.667 GET m.add_btn ;
PICTURE "@*HN \<Add" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('ADD') ;
MESSAGE 'Add a new record.'
@ 17.538,49.667 GET m.edit_btn ;
PICTURE "@*HN Ed\<it" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('EDIT') ;
MESSAGE 'Edit current record.'
@ 17.538,57.667 GET m.del_btn ;
PICTURE "@*HN \<Delete" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;

```

```

        VALID btn_val('DELETE') ;
        MESSAGE 'Delete current record.'
@ 17.538,65.667 GET m.prnt_btn ;
        PICTURE "@*HN P\<rint" ;
        SIZE 1.769,7.833,0.667 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        VALID btn_val('PRINT') ;
        MESSAGE 'Print report.'
@ 17.538,73.667 GET m.exit_btn ;
        PICTURE "@*HN \<Close" ;
        SIZE 1.769,7.833,0.667 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        VALID btn_val('EXIT') ;
        MESSAGE 'Close screen.'

```

```

IF NOT WVISIBLE("_sat0xy4uf")
    ACTIVATE WINDOW _sat0xy4uf
ENDIF

```

```

* *****
*
* *   WindowsREAD contains clauses from SCREEN s7019883
* *
* *****
*

```

```

READ CYCLE ;
    ACTIVATE READACT() ;
    DEACTIVATE READDEAC() ;
    NOLOCK

```

```

RELEASE WINDOW _sat0xy4uf

```

```

#REGION 0

```

```
SET READBORDER &rborder
```

```
IF m.talkstat = "ON"  
    SET TALK ON
```

```
ENDIF
```

```
IF m.compstat = "ON"  
    SET COMPATIBLE ON
```

```
ENDIF
```

```
* *****  
* *  
* *      ROOMS/Windows Cleanup Code  
* *  
* *****  
*
```

```
#REGION 1
```

```
SET DELETED &wzolddelete
```

```
SET FIELDS &wzfields
```

```
ON ERROR &wzolderror
```

```
ON KEY LABEL ESCAPE &wzoldesc
```

```
DO CASE
```

```
CASE _DOS AND SET('DISPLAY')='VGA25'
```

```
    @24,0 CLEAR TO 24,79
```

```
CASE _DOS AND SET('DISPLAY')='VGA50'
```

```
    @49,0 CLEAR TO 49,79
```

```
CASE _DOS
```

```
    @24,0 CLEAR TO 24,79
```

```
ENDCASE
```

```
****Procedures****
```

```
* *****  
* *  
* *      ROOMS/Windows Supporting Procedures and Functions  
* *  
* *****  
*
```

#REGION 1

PROCEDURE readdeac

IF isediting

 ACTIVATE WINDOW '_sat0xy4uf'

 WAIT WINDOW C_EDITS NOWAIT

ENDIF

IF !WVISIBLE(WOUTPUT())

 CLEAR READ

 RETURN .T.

ENDIF

RETURN .F.

PROCEDURE readact

IF !isediting

 SELECT (m.wzalias)

 SHOW GETS

ENDIF

DO REFRESH

RETURN

PROCEDURE wizerrorhandler

 * This very simple error handler is primarily intended

 * to trap for General field OLE errors which may occur

 * during editing from the MODIFY GENERAL window.

 WAIT WINDOW message()

RETURN

PROCEDURE printrec

 PRIVATE sOldError,wizfname,saverec,savearea,tmpcurs,tmpstr

 PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record

 STORE 1 TO p_recs,p_output

 STORE 0 TO prnt_btn

 STORE RECNO() TO saverec

 m.sOldError=ON('error')

 DO pdialog

 IF m.prnt_btn = 2

 RETURN

 ENDIF


```

IF !FILE(ALIAS()+'FRX')
  m.wizfname=SYS(2004)+'WIZARDS\'+WIZARD.APP
  IF !FILE(m.wizfname)
    ON ERROR *
    m.wizfname=LOCFILE('WIZARD.APP','APP',C_LOCWIZ)
    ON ERROR &sOldError
    IF !WIZARD.APP$UPPER(m.wizfname)
      WAIT WINDOW C_NOWIZ
      RETURN
    ENDIF
  ENDIF
WAIT WINDOW C_MAKEREPO NOWAIT
m.savearea=SELECT()
m.tmpcurs='_'+LEFT(SYS(3),7)
CREATE CURSOR (m.tmpcurs) (comment m)
m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)
INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)
SELECT (m.savearea)
DO (m.wizfname) WITH
", 'WZ_QREPO', 'NOSCRN/CREATE', ALIAS(), m.tmpcurs
  USE IN (m.tmpcurs)
WAIT CLEAR
  IF !FILE(ALIAS()+'FRX') &&wizard could not create report
    WAIT WINDOW C_NOREPO
    RETURN
  ENDIF
ENDIF
ENDIF

m.pr_out=IIF(m.p_output=1, 'TO PRINT NOCONSOLE', 'PREVIEW')
m.pr_record=IIF(m.p_recs=1, 'NEXT 1', 'ALL')
REPORT FORM (ALIAS()) &pr_out &pr_record
GO m.saverec
RETURN

PROCEDURE BTN_VAL
  PARAMETER m.btnname
  DO CASE
  CASE m.btnname='TOP'
    GO TOP

```

```

        WAIT WINDOW C_TOPFILE NOWAIT
CASE m.btnname='PREV'
    IF !BOF()
        SKIP -1
    ENDIF
    IF BOF()
        WAIT WINDOW C_TOPFILE NOWAIT
        GO TOP
    ENDIF
CASE m.btnname='NEXT'
    IF !EOF()
        SKIP 1
    ENDIF
    IF EOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        GO BOTTOM
    ENDIF
CASE m.btnname='END'
    GO BOTTOM
    WAIT WINDOW C_ENDFILE NOWAIT
CASE m.btnname='LOCATE'
    DO loc_dlog
CASE m.btnname='ADD' AND !isediting &&add record
    isediting=.T.
    isadding=.T.
    =edithand('ADD')
    _curobj=1
    DO refresh
    SHOW GETS
    RETURN
CASE m.btnname='EDIT' AND !isediting &&edit record
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF RLOCK()
        isediting=.T.
        _curobj=1
        DO refresh
        RETURN

```

```

ELSE
    WAIT WINDOW C_NOLOCK
ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
    IF isadding
        =edithand('SAVE')
    ELSE
        GATHER MEMVAR MEMO
    ENDIF
    UNLOCK
    isediting=.F.
    isadding=.F.
    DO refresh
CASE m.btnname='DELETE' AND isediting &&cancel record
    IF isadding
        =edithand('CANCEL')
    ENDIF
    isediting=.F.
    isadding=.F.
    UNLOCK
    WAIT WINDOW C_ECANCEL NOWAIT
    DO refresh
CASE m.btnname='DELETE'
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF fox_alert(C_DELREC)
        DELETE
        IF !EOF() AND DELETED()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C_ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    ENDIF
CASE m.btnname='PRINT'
    DO printrec
    RETURN

```

```

CASE m.btnname='EXIT'
    m.bailout=.T.      &&this is needed if used with FoxApp
    CLEAR READ
    RETURN
ENDCASE
SCATTER MEMVAR MEMO
SHOW GETS
RETURN

```

PROCEDURE REFRESH

```

DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting
    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE
    SHOW GET loc_btn DISABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE
    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "\<Save"
    SHOW GET del_btn,1 PROMPT "\<Cancel"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
    RETURN
OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"
    SHOW GET del_btn,1 PROMPT "\<Delete"

```



```

        SHOW GETS ENABLE
    ENDCASE
    IF m.is2table
        SHOW GET add_btn DISABLE
    ENDIF
    ON KEY LABEL ESCAPE
RETURN

```

```

PROCEDURE edithand
    PARAMETER m.paction
    * procedure handles edits
    DO CASE
    CASE m.paction = 'ADD'
        SCATTER MEMVAR MEMO BLANK
    CASE m.paction = 'SAVE'
        INSERT INTO (ALIAS()) FROM MEMVAR
    CASE m.paction = 'CANCEL'
        * nothing here
    ENDCASE
RETURN

```

```

PROCEDURE fox_alert
    PARAMETER wzalrtmess
    PRIVATE alrtbtn
    m.alrtbtn=2
    DEFINE WINDOW _qec1ij2t7 AT 0,0 SIZE 8,50 ;
    FONT "MS Sans Serif",10 STYLE 'B' ;
    FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()
    MOVE WINDOW _qec1ij2t7 CENTER
    ACTIVATE WINDOW _qec1ij2t7 NOSHOW
    @ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;
    FONT "MS Sans Serif", 10 STYLE "B"
    @ 6,18 GET m.alrtbtn ;
    PICTURE "@*HT \<OK;?\!\<Cancel" ;
    SIZE 1.769,8.667,1.333 ;
    FONT "MS Sans Serif", 8 STYLE "B"
    ACTIVATE WINDOW _qec1ij2t7
    READ CYCLE MODAL
    RELEASE WINDOW _qec1ij2t7

```

RETURN m.alrtbtn=1

PROCEDURE pdialog

```
    DEFINE WINDOW _qjn12zbvh ;  
        AT 0.000, 0.000 ;  
        SIZE 13.231, 54.800 ;  
        TITLE "Microsoft FoxPro" ;  
        FONT "MS Sans Serif", 8 ;  
        FLOAT NOCLOSE MINIMIZE SYSTEM  
MOVE WINDOW _qjn12zbvh CENTER  
ACTIVATE WINDOW _qjn12zbvh NOSHOW  
@ 2.846, 33.600 SAY "Output:" ;  
    FONT "MS Sans Serif", 8 ;  
    STYLE "BT"  
@ 2.846, 4.800 SAY "Print:" ;  
    FONT "MS Sans Serif", 8 ;  
    STYLE "BT"  
@ 4.692, 7.200 GET m.p_recs ;  
    PICTURE "@*RVN \<Current Record;\<All Records" ;  
    SIZE 1.308, 18.500, 0.308 ;  
    DEFAULT 1 ;  
    FONT "MS Sans Serif", 8 ;  
    STYLE "BT"  
@ 4.692, 36.000 GET m.p_output ;  
    PICTURE "@*RVN \<Printer;Pre\<view" ;  
    SIZE 1.308, 12.000, 0.308 ;  
    DEFAULT 1 ;  
    FONT "MS Sans Serif", 8 ;  
    STYLE "BT"  
@ 10.154, 16.600 GET m.prnt_btn ;  
    PICTURE "@*HT P\<rint;Ca\<ncel" ;  
    SIZE 1.769, 8.667, 0.667 ;  
    DEFAULT 1 ;  
    FONT "MS Sans Serif", 8 ;  
    STYLE "B"  
ACTIVATE WINDOW _qjn12zbvh  
READ CYCLE MODAL  
RELEASE WINDOW _qjn12zbvh
```

RETURN

```

PROCEDURE loc_dlog
  PRIVATE gfields,i
  DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;
    SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans
Serif",8
  MOVE WINDOW wzlocate CENTER
  m.gfields=SET('FIELDS',2)
  IF !EMPTY(RELATION(1))
    SET FIELDS ON
    IF m.gfields # 'GLOBAL'
      SET FIELDS GLOBAL
    ENDIF
    IF EMPTY(FLDLIST())
      m.i=1
      DO WHILE !EMPTY(OBJVAR(m.i))
        IF ATC('M.',OBJVAR(m.i))=0
          SET FIELDS TO (OBJVAR(m.i))
        ENDIF
        m.i = m.i + 1
      ENDDO
    ENDIF
  ENDIF
  BROWSE WINDOW wzlocate NOEDIT NODELETE ;
    NOMENU TITLE C_BRTITLE
  SET FIELDS &gfields
  SET FIELDS OFF
  RELEASE WINDOW wzlocate
RETURN

```

5- RESERVED ROOMS UERY:

[illegible]

```
SELECT Register.name, Register.surname, Register.sex, Register.age,;
Register.nation, Register.pass_no, Register.occupation, Register.address,;
Register.arr_date, Register.dep_date, Register.room_no;
FROM Register;
WHERE Register.name = "R"
```


EMPTY ROOMS QUERY:

[illegible]

```
SELECT Register.name, Register.surname, Register.sex, Register.age,;
Register.nation, Register.pass_no, Register.occupation, Register.address,;
Register.arr_date, Register.dep_date, Register.room_no;
FROM Register;
WHERE Register.name = "E"
```


7- TEL_LIST SCREEN:



Manager:	124
V.President:	763
Reception:	731
Registration:	831
Reservation:	731
Security:	837
Restaurant:	183
Accountant:	713
Employees:	183
Sports:	713
Supermark:	818

Edit **Print** **Close**

```

* *****
*
* * 05/04/00          PHONE.PRG          22:16:26
*
* *****
*
* * Author's Name
*
* * Copyright (c) 2000 Company Name
* * Address
* * City, Zip
*
* * Description:
* * This program was automatically generated by GENSCRN.
*
* *****

* *****
*
* *          PHONE/Windows Setup Code - SECTION 1
*
* *****
*

```

```

#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF

```

```

m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"
    SET TALK ON
ENDIF

```

```

#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat

```

```

IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS

```

```

m.rborder = SET("READBORDER")
SET READBORDER ON

```

```

m.currarea = SELECT()

```

```

* *****
* *
* *   PHONE/Windows Databases, Indexes, Relations
* *
* *****
*

```

```

IF USED("phone")
    SELECT phone
    SET ORDER TO 0
ELSE
    SELECT 0
    USE (LOCFILE("phone.dbf","DBF","Where is phone?"));
    AGAIN ALIAS phone ;

```

```

ORDER 0
ENDIF

```

```

* *****
*
*      Windows Window definitions
*
* *****
*

```

```

IF NOT WEXIST("_02o1bqozq")
    DEFINE WINDOW _02o1bqozq ;
        AT 0.000, 0.000 ;
        SIZE 31.769, 67.600 ;
        TITLE "Cyprus Hotel" ;
        FONT "Times New Roman", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        COLOR RGB(,,128,0,0)
    MOVE WINDOW _02o1bqozq CENTER
ENDIF

```

```

* *****
*
*      PHONE/Windows Setup Code - SECTION 2
*
* *****
*

```

```

#REGION 1

```

```

#DEFINE C_DBFEMPTY      'Database is empty, add a record?'
#DEFINE C_EDITS          'Please finish your edits.'
#DEFINE C_TOPFILE        'Top of file.'
#DEFINE C_ENDFILE        'End of file.'
#DEFINE C_BRTITLE        'Locate Record'

```


#DEFINE C_NOLOCK	'Sorry, could not lock record -- try again later.'
#DEFINE C_ECANCEL	'Edits Canceled.'
#DEFINE C_DELREC	'Delete selected record?'
#DEFINE C_NOFEAT	'Feature not available yet.'
#DEFINE C_NOWIZ	'Wizard application is not available.'
#DEFINE C_MAKEREPO	'Creating report with Report Wizard.'
#DEFINE C_NOREPO	'Could not create report.'
#DEFINE C_DELNOTE	'Deleting records...'
#DEFINE C_READONLY	'Table is read-only. No editing allowed.'
#DEFINE C_NOTABLE	'No table selected. Open table or run query.'
#DEFINE C_BADEXP	'Invalid expression.'
#DEFINE C_LOCWIZ	'Locate WIZARD.APP.'
#DEFINE C_MULTITABLE	'You have multiple related tables. Adding records in not allowed.'

```

MOVE WINDOW '_02o1bqozq' CENTER
PRIVATE isediting,isadding,wztblarr
PRIVATE wzolddelete,wzolderror,wzoldesc
PRIVATE wzalias, tempcurs,wzlastrec
PRIVATE isreadonly,find_drop,is2table

```

```

IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF

```

```

m.wztblarr= "
m.wzalias=SELECT()
m.isediting=.F.
m.isadding=.F.
m.is2table = .F.
m.wzolddelete=SET('DELETE')
SET DELETED ON
m.tempcurs=SYS(2015) &&used if General field
m.wzlastrec = 1
m.wzolderror=ON('error')
ON ERROR DO wizerrorhandler
wzoldesc=ON('KEY','ESCAPE')
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)

```


m.isreadonly=IIF(ISREAD(),.T.,.F.)

IF m.isreadonly

 WAIT WINDOW C_READONLY TIMEOUT 1

ENDIF

IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)

 APPEND BLANK

ENDIF

GOTO TOP

SCATTER MEMVAR MEMO

```
* *****
*
*      PHONE/Windows Screen Layout
*
* *****
*
```

#REGION 1

IF WVISIBLE("_02o1bqozq")

 ACTIVATE WINDOW _02o1bqozq SAME

ELSE

 ACTIVATE WINDOW _02o1bqozq NOSHOW

ENDIF

@ 1.538,17.400 SAY "CYPRUS HOTEL" ;

 FONT "Times New Roman", 16 ;

 STYLE "BT" ;

 COLOR RGB(255,255,255,,,) ;

@ 5.769,21.400 SAY "Manager:" ;

 SIZE 1.000,13.200 ;

 FONT "Times New Roman", 8 ;

 STYLE "BT" ;

 PICTURE "@J" ;

 COLOR RGB(255,255,255,255,255,255) ;

@ 5.769,41.200 GET m.manager ;

 SIZE 1.000,5.200 ;

 DEFAULT " " ;

FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 7.615,21.400 SAY "V.President:" ;
 SIZE 1.000,13.200 ;
 FONT "Times New Roman", 8 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(255,255,255,255,255,255)
 @ 7.615,41.200 GET m.vice ;
 SIZE 1.000,5.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 9.462,21.400 SAY "Reception:" ;
 SIZE 1.000,13.200 ;
 FONT "Times New Roman", 8 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(255,255,255,255,255,255)
 @ 9.462,41.200 GET m.reception ;
 SIZE 1.000,5.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 11.308,19.800 SAY "Registration:" ;
 SIZE 1.000,15.000 ;
 FONT "Times New Roman", 8 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(255,255,255,255,255,255)
 @ 11.308,41.200 GET m.register ;

SIZE 1.000,5.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 13.154,21.400 SAY "Reservation:" ;
 SIZE 1.000,13.200 ;
 FONT "Times New Roman", 8 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(255,255,255,255,255,255)
 @ 13.154,41.200 GET m.reserv ;
 SIZE 1.000,5.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 15.000,21.400 SAY "Security:" ;
 SIZE 1.000,13.200 ;
 FONT "Times New Roman", 8 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(255,255,255,255,255,255)
 @ 15.000,41.200 GET m.security ;
 SIZE 1.000,5.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 16.846,21.400 SAY "Restaurant:" ;
 SIZE 1.000,13.200 ;
 FONT "Times New Roman", 8 ;
 STYLE "BT" ;
 PICTURE "@J" ;

COLOR RGB(255,255,255,255,255,255)

@ 16.846,41.200 GET m.restaurant ;
 SIZE 1.000,5.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)

@ 18.692,21.400 SAY "Accountant:" ;
 SIZE 1.000,13.200 ;
 FONT "Times New Roman", 8 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(255,255,255,255,255,255)

@ 18.692,41.200 GET m.account ;
 SIZE 1.000,5.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)

@ 20.538,21.400 SAY "Employees:" ;
 SIZE 1.000,13.200 ;
 FONT "Times New Roman", 8 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(255,255,255,255,255,255)

@ 20.538,41.200 GET m.employees ;
 SIZE 1.000,5.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)

@ 22.385,21.400 SAY "Sports:" ;
 SIZE 1.000,13.200 ;
 FONT "Times New Roman", 8 ;


```

STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(255,255,255,255,255,255)
@ 22.385,41.200 GET m.sports ;
SIZE 1.000,5.200 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K 999" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 24.231,20.200 SAY "Supermark:" ;
SIZE 1.000,14.400 ;
FONT "Times New Roman", 8 ;
STYLE "BT" ;
PICTURE "@J" ;
COLOR RGB(255,255,255,255,255,255)
@ 24.231,41.200 GET m.supermark ;
SIZE 1.000,5.200 ;
DEFAULT " " ;
FONT "Times New Roman", 8 ;
STYLE "B" ;
PICTURE "@K 999" ;
WHEN isediting ;
COLOR ,RGB(0,0,0,255,255,255)
@ 28.308,16.400 GET m.edit_btn ;
PICTURE "@*HN Ed\<it" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('EDIT') ;
MESSAGE 'Edit current record.' ;
COLOR ,,,,,,RGB(0,0,0,,,,)
@ 28.308,26.600 GET m.prnt_btn ;
PICTURE "@*HN P\<rint" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;

```



```

VALID btn_val('PRINT') ;
MESSAGE 'Print report.' ;
COLOR ,,,,,,RGB(0,0,0,,,,)
@ 28.308,37.200 GET m.exit_btn ;
PICTURE "@*HN \<Close" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('EXIT') ;
MESSAGE 'Close screen.' ;
COLOR ,,,,,,RGB(0,0,0,,,,)
@ 4.769,9.800 TO 26.154,54.800 ;
PEN 1, 8 ;
STYLE "16" ;
COLOR RGB(255,255,255,,,,)

IF NOT WVISIBLE("_02o1bqozq")
    ACTIVATE WINDOW _02o1bqozq
ENDIF

```

```

* *****
* *
* * WindowsREAD contains clauses from SCREEN phone
* *
* *****
*

```

```

READ CYCLE ;
    ACTIVATE READACT() ;
    DEACTIVATE READDEAC() ;
    NOLOCK

```

```

RELEASE WINDOW _02o1bqozq

```

```

* *****
* *
* * Windows Closing Databases
* *

```

```
* *****
*
```

```
IF USED("phone")
    SELECT phone
    USE
ENDIF
```

```
SELECT (m.currarea)
```

```
#REGION 0
```

```
SET READBORDER &rborder
```

```
IF m.talkstat = "ON"
    SET TALK ON
```

```
ENDIF
```

```
IF m.compstat = "ON"
    SET COMPATIBLE ON
```

```
ENDIF
```

```
* *****
*
*      PHONE/Windows Cleanup Code
*
* *****
*
```

```
#REGION 1
```

```
SET DELETED &wzolddelete
```

```
SET FIELDS &wzfields
```

```
ON ERROR &wzolderror
```

```
ON KEY LABEL ESCAPE &wzoldesc
```

```
DO CASE
```

```
CASE _DOS AND SET('DISPLAY')='VGA25'
    @24,0 CLEAR TO 24,79
```

```
CASE _DOS AND SET('DISPLAY')='VGA50'
    @49,0 CLEAR TO 49,79
```

```

CASE _DOS
    @24,0 CLEAR TO 24,79
ENDCASE

```

****Procedures****

```

* *****
* *
* * PHONE/Windows Supporting Procedures and Functions
* *
* *****
*

```

#REGION 1

PROCEDURE readdeac

```

IF isediting
    ACTIVATE WINDOW '_02o1bqozq'
    WAIT WINDOW C_EDITS NOWAIT
ENDIF
IF !WVISIBLE(WOUTPUT())
    CLEAR READ
    RETURN .T.
ENDIF
RETURN .F.

```

PROCEDURE readact

```

IF !isediting
    SELECT (m.wzalias)
    SHOW GETS
ENDIF
DO REFRESH
RETURN

```

PROCEDURE wizerrorhandler

```

* This very simple error handler is primarily intended
* to trap for General field OLE errors which may occur
* during editing from the MODIFY GENERAL window.
    WAIT WINDOW message()
RETURN

```

```

PROCEDURE printrec
    PRIVATE sOldError,wizfname,saverec,savearea,tmpcurs,tmpstr
    PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record
    STORE 1 TO p_recs,p_output
    STORE 0 TO prnt_btn
    STORE RECNO() TO saverec
    m.sOldError=ON('error')
    DO pdialog
    IF m.prnt_btn = 2
        RETURN
    ENDIF
    IF !FILE(ALIAS()+'FRX')
        m.wizfname=SYS(2004)+'WIZARDS\'+WIZARD.APP
        IF !FILE(m.wizfname)
            ON ERROR *
            m.wizfname=LOCFILE('WIZARD.APP','APP',C_LOCWIZ)
            ON ERROR &sOldError
            IF !WIZARD.APP$UPPER(m.wizfname)
                WAIT WINDOW C_NOWIZ
                RETURN
            ENDIF
        ENDIF
    ENDIF
    WAIT WINDOW C_MAKEREPO NOWAIT
    m.savearea=SELECT()
    m.tmpcurs='_'+LEFT(SYS(3),7)
    CREATE CURSOR (m.tmpcurs) (comment m)
    m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)
    INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)
    SELECT (m.savearea)
    DO (m.wizfname) WITH
    ",WZ_QREPO',NOSCRN/CREATE',ALIAS(),m.tmpcurs
    USE IN (m.tmpcurs)
    WAIT CLEAR
    IF !FILE(ALIAS()+'FRX') &&wizard could not create report
        WAIT WINDOW C_NOREPO
        RETURN
    ENDIF
ENDIF

```

```

m.pr_out=IIF(m.p_output=1,'TO PRINT NOCONSOLE','PREVIEW')
m.pr_record=IIF(m.p_recs=1,'NEXT 1','ALL')
REPORT FORM (ALIAS()) &pr_out &pr_record
GO m.saverec
RETURN

```

```

PROCEDURE BTN_VAL
PARAMETER m.btnname
DO CASE
CASE m.btnname='TOP'
GO TOP
WAIT WINDOW C_TOPFILE NOWAIT
CASE m.btnname='PREV'
IF !BOF()
SKIP -1
ENDIF
IF BOF()
WAIT WINDOW C_TOPFILE NOWAIT
GO TOP
ENDIF
CASE m.btnname='NEXT'
IF !EOF()
SKIP 1
ENDIF
IF EOF()
WAIT WINDOW C_ENDFILE NOWAIT
GO BOTTOM
ENDIF
CASE m.btnname='END'
GO BOTTOM
WAIT WINDOW C_ENDFILE NOWAIT
CASE m.btnname='LOCATE'
DO loc_dlog
CASE m.btnname='ADD' AND !isediting &&add record
isediting=.T.
isadding=.T.
=edithand('ADD')
_curobj=1

```



```

DO refresh
SHOW GETS
RETURN
CASE m.btnname='EDIT' AND !isediting &&edit record
IF EOF() OR BOF()
    WAIT WINDOW C_ENDFILE NOWAIT
    RETURN
ENDIF
IF RLOCK()
    isediting=.T.
    _curobj=1
    DO refresh
    RETURN
ELSE
    WAIT WINDOW C_NOLOCK
ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
IF isadding
    =edithand('SAVE')
ELSE
    GATHER MEMVAR MEMO
ENDIF
UNLOCK
isediting=.F.
isadding=.F.
DO refresh
CASE m.btnname='DELETE' AND isediting &&cancel record
IF isadding
    =edithand('CANCEL')
ENDIF
isediting=.F.
isadding=.F.
UNLOCK
WAIT WINDOW C_ECANCEL NOWAIT
DO refresh
CASE m.btnname='DELETE'
IF EOF() OR BOF()
    WAIT WINDOW C_ENDFILE NOWAIT
    RETURN
ENDIF

```

```

        IF fox_alert(C_DELREC)
            DELETE
            IF !EOF() AND DELETED()
                SKIP 1
            ENDIF
            IF EOF()
                WAIT WINDOW C_ENDFILE NOWAIT
                GO BOTTOM
            ENDIF
        ENDIF
    CASE m.btnname='PRINT'
        DO printrec
        RETURN
    CASE m.btnname='EXIT'
        m.bailout=.T.      &&this is needed if used with FoxApp
        CLEAR READ
        RETURN
    ENDCASE
    SCATTER MEMVAR MEMO
    SHOW GETS
RETURN

```

PROCEDURE REFRESH

```

DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting
    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE

```

```

    SHOW GET loc_btn DISABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE
    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "\<Save"
    SHOW GET del_btn,1 PROMPT "\<Cancel"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
    RETURN
OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"
    SHOW GET del_btn,1 PROMPT "\<Delete"
    SHOW GETS ENABLE
ENDCASE
IF m.is2table
    SHOW GET add_btn DISABLE
ENDIF
ON KEY LABEL ESCAPE
RETURN

```

```

PROCEDURE edithand
    PARAMETER m.paction
    * procedure handles edits
    DO CASE
    CASE m.paction = 'ADD'
        SCATTER MEMVAR MEMO BLANK
    CASE m.paction = 'SAVE'
        INSERT INTO (ALIAS()) FROM MEMVAR
    CASE m.paction = 'CANCEL'
        * nothing here
    ENDCASE
RETURN

```

```

PROCEDURE fox_alert
    PARAMETER wzalrtmess
    PRIVATE alrtbtn
    m.alrtbtn=2
    DEFINE WINDOW _qec1ij2t7 AT 0,0 SIZE 8,50 ;

```

```

    FONT "MS Sans Serif",10 STYLE 'B' ;
    FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()
MOVE WINDOW _qec1ij2t7 CENTER
ACTIVATE WINDOW _qec1ij2t7 NOSHOW
@ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;
    FONT "MS Sans Serif", 10 STYLE "B"
@ 6,18 GET m.alrtbtn ;
    PICTURE "@*HT \<OK;\?! \<Cancel" ;
    SIZE 1.769,8.667,1.333 ;
    FONT "MS Sans Serif", 8 STYLE "B"
ACTIVATE WINDOW _qec1ij2t7
READ CYCLE MODAL
RELEASE WINDOW _qec1ij2t7
RETURN m.alrtbtn=1

```

PROCEDURE pdialog

```

    DEFINE WINDOW _qjn12zbvh ;
        AT 0.000, 0.000 ;
        SIZE 13.231,54.800 ;
        TITLE "Microsoft FoxPro" ;
        FONT "MS Sans Serif", 8 ;
        FLOAT NOCLOSE MINIMIZE SYSTEM
MOVE WINDOW _qjn12zbvh CENTER
ACTIVATE WINDOW _qjn12zbvh NOSHOW
@ 2.846,33.600 SAY "Output:" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 2.846,4.800 SAY "Print:" ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 4.692,7.200 GET m.p_recs ;
    PICTURE "@*RVN \<Current Record;\<All Records" ;
    SIZE 1.308,18.500,0.308 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "BT"
@ 4.692,36.000 GET m.p_output ;
    PICTURE "@*RVN \<Printer;Pre\<view" ;
    SIZE 1.308,12.000,0.308 ;

```



```

        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
    @ 10.154,16.600 GET m.prnt_btn ;
        PICTURE "@*HT P\<rint;Ca\<ncel" ;
        SIZE 1.769,8.667,0.667 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B"
    ACTIVATE WINDOW _qjn12zbvh
    READ CYCLE MODAL
    RELEASE WINDOW _qjn12zbvh
RETURN

PROCEDURE loc_dlog
    PRIVATE gfields,i
    DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;
        SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans
Serif",8
    MOVE WINDOW wzlocate CENTER
    m.gfields=SET('FIELDS',2)
    IF !EMPTY(RELATION(1))
        SET FIELDS ON
        IF m.gfields # 'GLOBAL'
            SET FIELDS GLOBAL
        ENDIF
        IF EMPTY(FLDLIST())
            m.i=1
            DO WHILE !EMPTY(OBJVAR(m.i))
                IF ATC('M.',OBJVAR(m.i))=0
                    SET FIELDS TO (OBJVAR(m.i))
                ENDIF
                m.i = m.i + 1
            ENDDO
        ENDIF
    ENDIF
    BROWSE WINDOW wzlocate NOEDIT NODELETE ;
        NOMENU TITLE C _BRTITLE
    SET FIELDS &gfields

```

4- CUSTOMER FILE:

Cyprus Hotel

WELCOME TO CYPRUS HOTEL

Name: Surname:

Sex: ☒ Male ☐ Female Age:

Nationality: Passport No:

Occupation:

Address:

Arrival date: Departure date:

Room No: Room Type:

Room Phone: Room Price: \$

```

* *****
*
* * 10.12.1999      REGISTER.SPR      15:50:17
*
* *****
*
* * Author's Name
*
* * Copyright (c) 1999 Company Name
* * Address
* * City, Zip
*
* * Description:
* * This program was automatically generated by GENSCRN.
*
* *****

* *****
*
* * REGISTER/Windows Setup Code - SECTION 1
*
* *****
*

```

```

#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF
m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"
    SET TALK ON
ENDIF

```

```
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
```

```
IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
ELSE
    m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
```

```
m.rborder = SET("READBORDER")
SET READBORDER ON
```

```
IF USED("register");
    SELECT register;
    ORDER tag ('_san0ormw1')
ELSE
    SELECT 0
    USE (LOCFILE("c:\taisir\register.dbf","DBF","Where is register?"));
    AGAIN ALIAS register ;
    ORDER tag ('_san0ormw1')
ENDIF
```

```
ENDIF
* *****
* *
* *      Windows Window definitions
* *
* *****
*
```

```
IF NOT WEXIST("_sat0xy38o")
    DEFINE WINDOW _sat0xy38o ;
        AT 0.000,0.000 ;
        SIZE 31.000,97.833 ;
        TITLE "Cyprus Hotel" ;
        FONT "MS Sans Serif", 8 ;
```



```

        STYLE "B" ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        COLOR RGB(,,128,0,0)
    MOVE WINDOW _sat0xy38o CENTER
ENDIF

```

```

*      *****
*      *
*      *      REGISTER/Windows Setup Code - SECTION 2
*      *
*      *****
*

```

```

#REGION 1

```

```

#DEFINE C_DBFEMPTY      'Database is empty, add a record?'
#DEFINE C_EDITS          'Please finish your edits.'
#DEFINE C_TOPFILE        'Top of file.'
#DEFINE C_ENDFILE        'End of file.'
#DEFINE C_BRTITLE        'Locate Record'
#DEFINE C_NOLOCK         'Sorry, could not lock record -- try again later.'
#DEFINE C_ECANCEL        'Edits Canceled.'
#DEFINE C_DELREC         'Delete selected record?'
#DEFINE C_NOFEAT         'Feature not available yet.'
#DEFINE C_NOWIZ          'Wizard application is not available.'
#DEFINE C_MAKEREPO       'Creating report with Report Wizard.'
#DEFINE C_NOREPO         'Could not create report.'
#DEFINE C_DELNOTE        'Deleting records...'
#DEFINE C_READONLY       'Table is read-only. No editing allowed.'
#DEFINE C_NOTABLE        'No table selected. Open table or run query.'
#DEFINE C_BADEXP         'Invalid expression.'
#DEFINE C_LOCWIZ         'Locate WIZARD.APP:'
#DEFINE C_MULTITABLE     'You have multiple related tables. Adding
records in not allowed.'

```

```

MOVE WINDOW '_sat0xy38o' CENTER
PRIVATE isediting,isadding,wztblarr

```

```
PRIVATE wzolddelete,wzolderror,wzoldesc
PRIVATE wzalias, tempcurs,wzlastrec
PRIVATE isreadonly,find_drop,is2table
```

```
IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF
```

```
m.wztblarr= "
m.wzalias=SELECT()
m.isediting=.F.
m.isadding=.F.
m.is2table = .F.
m.wzolddelete=SET('DELETE')
SET DELETED ON
m.tempcurs=SYS(2015) &&used if General field
m.wzlastrec = 1
m.wzolderror=ON('error')
ON ERROR DO wizerrorhandler
wzoldesc=ON('KEY','ESCAPE')
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)

m.isreadonly=IIF(ISREAD(),.T.,.F.)
IF m.isreadonly
    WAIT WINDOW C_READONLY TIMEOUT 1
ENDIF
```

```
IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)
    APPEND BLANK
ENDIF
```

```
GOTO TOP
SCATTER MEMVAR MEMO
```

```
* *****
* *
* * REGISTER/Windows Screen Layout
```

```
*
*
* *****
*
```

```
#REGION 1
IF WVISIBLE("_sat0xy38o")
    ACTIVATE WINDOW _sat0xy38o SAME
ELSE
    ACTIVATE WINDOW _sat0xy38o NOSHOW
ENDIF
@ 0.385,22.000 SAY "WELCOME TO CYPRUS HOTEL" ;
    FONT "MS Sans Serif", 14 ;
    STYLE "BT" ;
    COLOR RGB(255,255,255,,,)
@ 2.692,5.167 TO 2.692,88.500 ;
    PEN 2, 8 ;
    STYLE "I" ;
    COLOR RGB(255,255,255,,,)
@ 26.538,5.833 TO 26.538,89.166 ;
    PEN 2, 8 ;
    STYLE "I" ;
    COLOR RGB(255,255,255,,,)
@ 3.692,4.333 SAY "Name:" ;
    SIZE 0.938,15.600 ;
    FONT "Times New Roman", 9 ;
    STYLE "BT" ;
    PICTURE "@J" ;
    COLOR RGB(,,255,255,255)
@ 3.692,49.833 SAY "Surname:" ;
    SIZE 0.938,15.600 ;
    FONT "Times New Roman", 9 ;
    STYLE "BT" ;
    PICTURE "@J" ;
    COLOR RGB(,,255,255,255)
@ 3.769,20.000 GET m.name ;
    SIZE 1.000,31.000 ;
    DEFAULT " " ;
    FONT "Times New Roman", 8 ;
    STYLE "B" ;
    PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" ;
```

WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 3.769,65.500 GET m.surname ;
 SIZE 1.000,36.800 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 5.538,4.333 SAY "Sex:" ;
 SIZE 0.938,15.600 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,,255,255,255)
 @ 5.615,20.000 GET m.sex ;
 PICTURE "@*RHN Male;Female" ;
 SIZE 1.308,11.000,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "BT"
 @ 5.615,50.000 SAY "Age:" ;
 SIZE 0.938,15.600 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,,255,255,255)
 @ 5.692,65.667 GET m.age ;
 SIZE 1.000,2.400 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 99" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 7.385,4.333 SAY "Nationonality:" ;
 SIZE 0.938,15.600 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;

PICTURE "@J" ;
 COLOR RGB(,,,255,255,255)
 @ 7.385,49.833 SAY "Passport No:" ;
 SIZE 0.938,15.600 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,,255,255,255)
 @ 7.462,20.000 GET m.nation ;
 SIZE 1.000,23.200 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 7.462,65.500 GET m.pass_no ;
 SIZE 1.000,16.000 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K XXXXXXXXXXXX" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 9.231,4.333 SAY "Occupation:" ;
 SIZE 0.938,15.600 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,,255,255,255)
 @ 9.308,20.000 GET m.occupation ;
 SIZE 1.000,41.400 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 11.077,4.333 SAY "Address:" ;
 SIZE 0.938,15.600 ;

FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,255,255,255)
 @ 11.154,20.333 EDIT M.ADDRESS ;
 SIZE 6.000,48.333,120.000 ;
 PICTURE "@K
 XX" ;
 DEFAULT " " ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 SCROLL
 @ 18.462,4.333 SAY "Arrival date:" ;
 SIZE 0.938,15.600 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,255,255,255)
 @ 18.462,49.667 SAY "Departure date:" ;
 SIZE 0.938,20.800 ;
 @ FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,255,255,255)
 @ 18.538,20.000 GET m.arr_date ;
 SIZE 1.000,12.800 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 999999999999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 18.538,70.333 GET m.dep_date ;
 SIZE 1.000,11.400 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)

@ 20.308,4.333 SAY "Room No:" ;
 SIZE 0.938,15.600 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,255,255,255)
 @ 20.308,49.667 SAY "Room Type:" ;
 SIZE 0.938,15.600 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,255,255,255)
 @ 20.385,20.000 GET m.room_no ;
 SIZE 1.000,8.800 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K 9,999,999,999" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 20.385,65.333 GET m.room_type ;
 SIZE 1.000,16.000 ;
 DEFAULT " " ;
 FONT "Times New Roman", 8 ;
 STYLE "B" ;
 PICTURE "@K XXXXXXXXXXXX" ;
 WHEN isediting ;
 COLOR ,RGB(0,0,0,255,255,255)
 @ 22.154,4.333 SAY "Room Phone:" ;
 SIZE 0.938,15.600 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,255,255,255)
 @ 22.154,49.667 SAY "Room Price:" ;
 SIZE 0.938,15.600 ;
 FONT "Times New Roman", 9 ;
 STYLE "BT" ;
 PICTURE "@J" ;
 COLOR RGB(,,255,255,255)

```

@ 22.231,20.000 GET m.phone_no ;
    SIZE 1.000,6.800 ;
    DEFAULT " " ;
    FONT "Times New Roman", 8 ;
    STYLE "B" ;
    PICTURE "@K 999" ;
    WHEN isediting ;
    COLOR ,RGB(0,0,0,255,255,255)
@ 22.231,65.333 GET m.room_price ;
    SIZE 1.000,14.400 ;
    DEFAULT " " ;
    FONT "Times New Roman", 8 ;
    STYLE "B" ;
    PICTURE "@K 9999999999" ;
    WHEN isediting ;
    COLOR ,RGB(0,0,0,255,255,255)
@ 22.231,79.833 GET m.MONEY ;
    SIZE 1.000,2.400 ;
    DEFAULT " " ;
    FONT "Times New Roman", 8 ;
    STYLE "B" ;
    PICTURE "@K 99" ;
    WHEN isediting ;
    COLOR ,RGB(0,0,0,255,255,255)
@ 24.154,20.167 GET wzsshellpa ;
    PICTURE "@*HN Payment" ;
    SIZE 1.769,17.000,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID _sat0xy3c3()
@ 24.231,51.500 GET wzsshell ;
    PICTURE "@*HN Family" ;
    SIZE 1.769,16.333,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID _sat0xy3c9()
@ 27.692,7.500 GET m.top_btn ;
    PICTURE "@*HN \<Top" ;

```



```

SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('TOP') ;
MESSAGE 'Go to first record.'
@ 27.692,15.500 GET m.prev_btn ;
PICTURE "@*HN \<Prev" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('PREV') ;
MESSAGE 'Go to previous record.'
@ 27.692,23.500 GET m.next_btn ;
PICTURE "@*HN \<Next" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('NEXT') ;
MESSAGE 'Go to next record.'
@ 27.692,31.500 GET m.end_btn ;
PICTURE "@*HN \<End" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('END') ;
MESSAGE 'Go to last record.'
@ 27.692,39.500 GET m.loc_btn ;
PICTURE "@*HN \<Locate" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('LOCATE') ;
MESSAGE 'Locate a record.'
@ 27.692,47.500 GET m.add_btn ;
PICTURE "@*HN \<Add" ;

```

```

SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('ADD') ;
MESSAGE 'Add a new record.'
@ 27.692,55.500 GET m.edit_btn ;
PICTURE "@*HN Ed\<it" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('EDIT') ;
MESSAGE 'Edit current record.'
@ 27.692,63.500 GET m.del_btn ;
PICTURE "@*HN \<Delete" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('DELETE') ;
MESSAGE 'Delete current record.'
@ 27.692,71.500 GET m.prnt_btn ;
PICTURE "@*HN P\<rint" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('PRINT') ;
MESSAGE 'Print report.'
@ 27.692,79.500 GET m.exit_btn ;
PICTURE "@*HN \<Close" ;
SIZE 1.769,7.833,0.667 ;
DEFAULT 1 ;
FONT "MS Sans Serif", 8 ;
STYLE "B" ;
VALID btn_val('EXIT') ;
MESSAGE 'Close screen.'

IF NOT WVISIBLE("_sat0xy38o")

```

```
        ACTIVATE WINDOW _sat0xy38o
ENDIF
```

```
* *****
*
* *   WindowsREAD contains clauses from SCREEN register
* *
* *****
*
```

```
READ CYCLE ;
    ACTIVATE READACT() ;
    DEACTIVATE READDEAC() ;
    NOLOCK
```

```
RELEASE WINDOW _sat0xy38o
```

```
#REGION 0
```

```
SET READBORDER &rborder
```

```
IF m.talkstat = "ON"
    SET TALK ON
ENDIF
IF m.compstat = "ON"
    SET COMPATIBLE ON
ENDIF
```

```
* *****
*
* *   REGISTER/Windows Cleanup Code
* *
* *****
*
```

```
#REGION 1
SET DELETED &wzolddelete
SET FIELDS &wzfields
```

```

ON ERROR &wzolderror
ON KEY LABEL ESCAPE &wzoldesc
DO CASE
CASE _DOS AND SET('DISPLAY')='VGA25'
    @24,0 CLEAR TO 24,79
CASE _DOS AND SET('DISPLAY')='VGA50'
    @49,0 CLEAR TO 49,79
CASE _DOS
    @24,0 CLEAR TO 24,79
ENDCASE

```

****Procedures****

```

* *****
*
* * REGISTER/Windows Supporting Procedures and Functions
*
* *****
*

```

```

#REGION 1
PROCEDURE readdeac
    IF isediting
        ACTIVATE WINDOW '_sat0xy38o'
        WAIT WINDOW C_EDITS NOWAIT
    ENDIF
    IF !WVISIBLE(WOUTPUT())
        CLEAR READ
        RETURN .T.
    ENDIF
    RETURN .F.

```

```

PROCEDURE readact
    IF !isediting
        SELECT (m.wzalias)
        SHOW GETS
    ENDIF
    DO REFRESH
    RETURN

```


PROCEDURE wizerrorhandler

- * This very simple error handler is primarily intended
- * to trap for General field OLE errors which may occur
- * during editing from the MODIFY GENERAL window.

WAIT WINDOW message()

RETURN

PROCEDURE printrec

PRIVATE sOldError,wizfname,saverec,savearea,tmpcurs,tmpstr

PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record

STORE 1 TO p_recs,p_output

STORE 0 TO prnt_btn

STORE RECNO() TO saverec

m.sOldError=ON('error')

DO pdialog

IF m.prnt_btn = 2

RETURN

ENDIF

IF !FILE(ALIAS()+'FRX')

m.wizfname=SYS(2004)+'WIZARDS\'+'WIZARD.APP'

IF !FILE(m.wizfname)

ON ERROR *

m.wizfname=LOCFILE('WIZARD.APP','APP',C_LOCWIZ)

ON ERROR &sOldError

IF !'WIZARD.APP'\$UPPER(m.wizfname)

WAIT WINDOW C_NOWIZ

RETURN

ENDIF

ENDIF

WAIT WINDOW C_MAKEREPO NOWAIT

m.savearea=SELECT()

m.tmpcurs='_'+LEFT(SYS(3),7)

CREATE CURSOR (m.tmpcurs) (comment m)

m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)

INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)

SELECT (m.savearea)

DO (m.wizfname) WITH

", 'WZ_QREPO', 'NOSCRN/CREATE', ALIAS(), m.tmpcurs

```

        USE IN (m.tmpcurs)
WAIT CLEAR
        IF !FILE(ALIAS()+'.FRX') &&wizard could not create report
        WAIT WINDOW C_NOREPO
        RETURN
        ENDIF
    ENDIF

    m.pr_out=IIF(m.p_output=1,'TO PRINT NOCONSOLE','PREVIEW')
    m.pr_record=IIF(m.p_recs=1,'NEXT 1','ALL')
    REPORT FORM (ALIAS()) &pr_out &pr_record
    GO m.saverec
RETURN

```

```

PROCEDURE BTN_VAL
    PARAMETER m.btnname
    DO CASE
    CASE m.btnname='TOP'
        GO TOP
        WAIT WINDOW C_TOPFILE NOWAIT
    CASE m.btnname='PREV'
        IF !BOF()
            SKIP -1
        ENDIF
        IF BOF()
            WAIT WINDOW C_TOPFILE NOWAIT
            GO TOP
        ENDIF
    CASE m.btnname='NEXT'
        IF !EOF()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C_ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    CASE m.btnname='END'
        GO BOTTOM
        WAIT WINDOW C_ENDFILE NOWAIT
    
```

```

CASE m.btnname='LOCATE'
    DO loc_dlog
CASE m.btnname='ADD' AND !isediting &&add record
    isediting=.T.
    isadding=.T.
    =edithand('ADD')
    _curobj=1
    DO refresh
    SHOW GETS
    RETURN
CASE m.btnname='EDIT' AND !isediting &&edit record
    IF EOF() OR BOF()
        WAIT WINDOW C__ENDFILE NOWAIT
        RETURN
    ENDIF
    IF RLOCK()
        isediting=.T.
        _curobj=1
        DO refresh
        RETURN
    ELSE
        WAIT WINDOW C__NOLOCK
    ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
    IF isadding
        =edithand('SAVE')
    ELSE
        GATHER MEMVAR MEMO
    ENDIF
    UNLOCK
    isediting=.F.
    isadding=.F.
    DO refresh
CASE m.btnname='DELETE' AND isediting &&cancel record
    IF isadding
        =edithand('CANCEL')
    ENDIF
    isediting=.F.
    isadding=.F.
    UNLOCK

```

```

        WAIT WINDOW C_ECANCEL NOWAIT
        DO refresh
    CASE m.btnname='DELETE'
        IF EOF() OR BOF()
            WAIT WINDOW C_ENDFILE NOWAIT
            RETURN
        ENDIF
        IF fox_alert(C_DELREC)
            DELETE
            IF !EOF() AND DELETED()
                SKIP 1
            ENDIF
            IF EOF()
                WAIT WINDOW C_ENDFILE NOWAIT
                GO BOTTOM
            ENDIF
        ENDIF
    CASE m.btnname='PRINT'
        DO printrec
        RETURN
    CASE m.btnname='EXIT'
        m.bailout=.T.      &&this is needed if used with FoxApp
        CLEAR READ
        RETURN
    ENDCASE
    SCATTER MEMVAR MEMO
    SHOW GETS
RETURN

```

PROCEDURE REFRESH

```

DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting

```



```

    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE
    SHOW GET loc_btn DISABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE
    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "\<Save"
    SHOW GET del_btn,1 PROMPT "\<Cancel"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
    RETURN
OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"
    SHOW GET del_btn,1 PROMPT "\<Delete"
    SHOW GETS ENABLE
ENDCASE
IF m.is2table
    SHOW GET add_btn DISABLE
ENDIF
ON KEY LABEL ESCAPE
RETURN

```

```

PROCEDURE edithand
    PARAMETER m.paction
    * procedure handles edits
    DO CASE
    CASE m.paction = 'ADD'
        SCATTER MEMVAR MEMO BLANK
    CASE m.paction = 'SAVE'
        INSERT INTO (ALIAS()) FROM MEMVAR
    CASE m.paction = 'CANCEL'
        * nothing here
    ENDCASE

```

RETURN

PROCEDURE fox_alert

PARAMETER wzalrtmess

PRIVATE alrtbtn

m.alrtbtn=2

DEFINE WINDOW _qec1ij2t7 AT 0,0 SIZE 8,50 ;

FONT "MS Sans Serif",10 STYLE 'B' ;

FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()

MOVE WINDOW _qec1ij2t7 CENTER

ACTIVATE WINDOW _qec1ij2t7 NOSHOW

@ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;

FONT "MS Sans Serif", 10 STYLE "B"

@ 6,18 GET m.alrtbtn ;

PICTURE "@*HT \<OK;\?!\<Cancel" ;

SIZE 1.769,8.667,1.333 ;

FONT "MS Sans Serif", 8 STYLE "B"

ACTIVATE WINDOW _qec1ij2t7

READ CYCLE MODAL

RELEASE WINDOW _qec1ij2t7

RETURN m.alrtbtn=1

PROCEDURE pdialog

DEFINE WINDOW _qjn12zbvh ;

AT 0.000, 0.000 ;

SIZE 13.231,54.800 ;

TITLE "Microsoft FoxPro" ;

FONT "MS Sans Serif", 8 ;

FLOAT NOCLOSE MINIMIZE SYSTEM

MOVE WINDOW _qjn12zbvh CENTER

ACTIVATE WINDOW _qjn12zbvh NOSHOW

@ 2.846,33.600 SAY "Output:" ;

FONT "MS Sans Serif", 8 ;

STYLE "BT"

@ 2.846,4.800 SAY "Print:" ;

FONT "MS Sans Serif", 8 ;

STYLE "BT"

@ 4.692,7.200 GET m.p_recs ;

PICTURE "@*RVN \<Current Record;\<All Records" ;

```

        SIZE 1.308,18.500,0.308 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
    @ 4.692,36.000 GET m.p_output ;
        PICTURE "@*RVN \<Printer;Pre\<view" ;
        SIZE 1.308,12.000,0.308 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
    @ 10.154,16.600 GET m.pnt_btn ;
        PICTURE "@*HT P\<rint;Ca\<ncel" ;
        SIZE 1.769,8.667,0.667 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B"
    ACTIVATE WINDOW _qjn12zbvh
    READ CYCLE MODAL
    RELEASE WINDOW _qjn12zbvh
RETURN

PROCEDURE loc_dlog
    PRIVATE gfields,i
    DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;
        SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans
Serif",8
    MOVE WINDOW wzlocate CENTER
    m.gfields=SET('FIELDS',2)
    IF !EMPTY(RELATION(1))
        SET FIELDS ON
        IF m.gfields # 'GLOBAL'
            SET FIELDS GLOBAL
        ENDIF
    IF EMPTY(FLDLIST())
        m.i=1
        DO WHILE !EMPTY(OBJVAR(m.i))
            IF ATC('M.',OBJVAR(m.i))=0
                SET FIELDS TO (OBJVAR(m.i))
            ENDIF
        ENDIF
    ENDIF

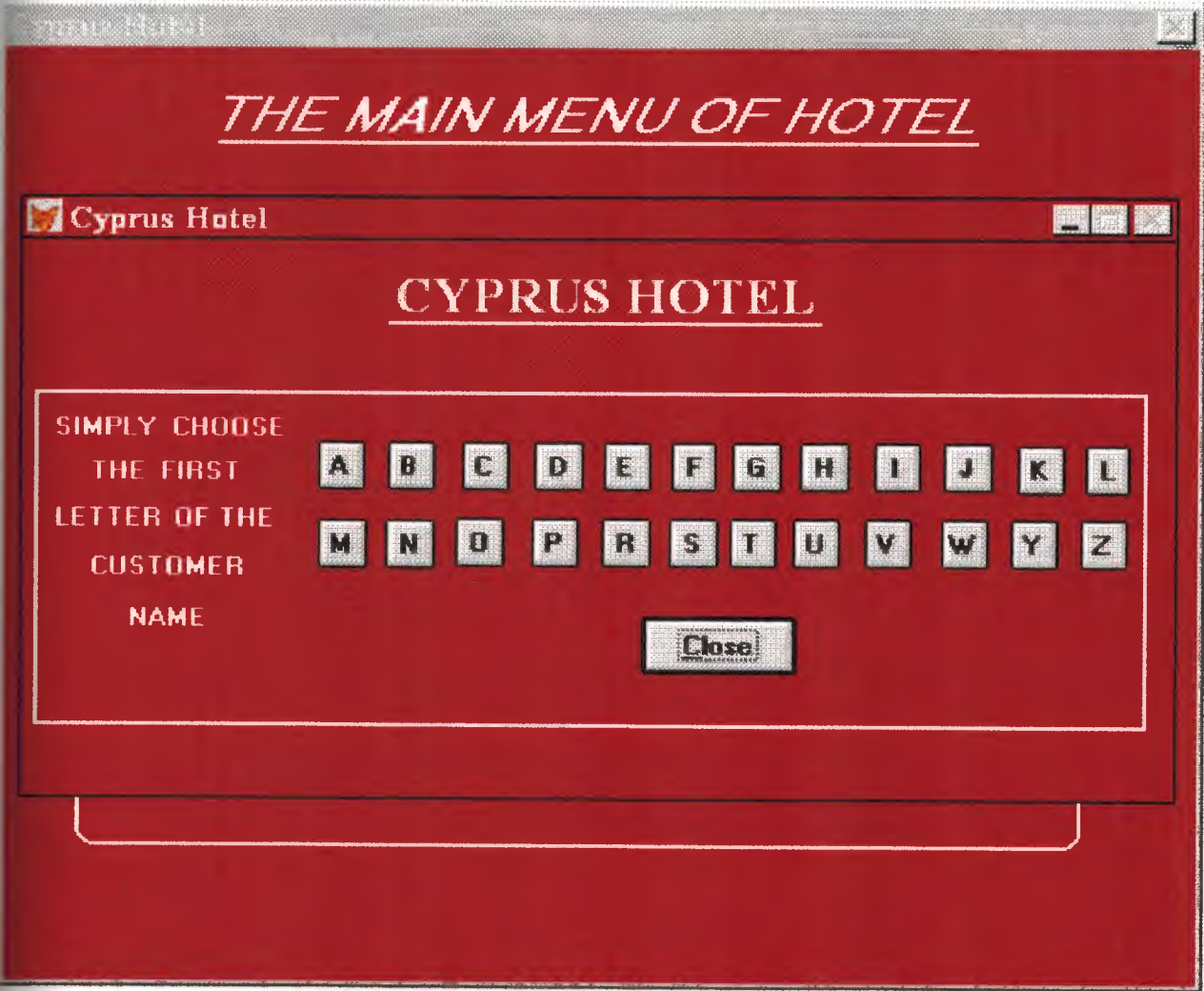
```

```

                                m.i = m.i + 1
                                ENDDO
                                ENDIF
                                ENDIF
                                BROWSE WINDOW wzlocate NOEDIT NODELETE ;
                                NOMENU TITLE C_BRTITLE
                                SET FIELDS &gfields
                                SET FIELDS OFF
                                RELEASE WINDOW wzlocate
RETURN
*   * _SAT0XY3C3          wzsshellpa VALID
*   * Function Origin:
*   * From Platform:      Windows
*   * From Screen:        REGISTER,   Record Number: 35
*   * Variable:           wzsshellpa
*   * Called By:          VALID Clause
*   * Object Type:        Push Button
*   * Snippet Number:     1
*   *
*   *****
*
FUNCTION _sat0xy3c3   && wzsshellpa VALID
#REGION 1
do c:/taisir/payment.prg
*   * _SAT0XY3C9          wzsshell VALID
*   * Function Origin:
*   * From Platform:      Windows
*   * From Screen:        REGISTER,   Record Number: 36
*   * Variable:           wzsshell
*   * Called By:          VALID Clause
*   * Object Type:        Push Button
*   * Snippet Number:     2
FUNCTION _sat0xy3c9   && wzsshell VALID
#REGION 1
do c:/taisir/family.prg

```


9- SEARCHING MENU:



```

* *****
*
* * 05/04/00      SEARCH.PRG      22:39:26
*
* *****
*
* * Author's Name
*
* * Copyright (c) 2000 Company Name
* * Address
* * City,  Zip
*
* * Description:
* * This program was automatically generated by GENSCRN.
*
* *****

```

```

* *****
*
* *      LOOKING/Windows Setup Code - SECTION 1
*
* *****
*

```

```

#REGION 1
PRIVATE wzfields,wztalk
IF SET("TALK") = "ON"
    SET TALK OFF
    m.wztalk = "ON"
ELSE
    m.wztalk = "OFF"
ENDIF
m.wzfields=SET('FIELDS')
SET FIELDS OFF
IF m.wztalk = "ON"

```

```
        SET TALK ON
ENDIF
```

```
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
```

```
IF SET("TALK") = "ON"
    SET TALK OFF
    m.talkstat = "ON"
```

```
ELSE
    m.talkstat = "OFF"
```

```
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
```

```
m.rborder = SET("READBORDER")
SET READBORDER ON
```

```
m.currarea = SELECT()
```

```
* *****
*
* *   LOOKING/Windows Databases, Indexes, Relations
*
* *****
*
```

```
IF USED("looking")
    SELECT looking
    SET ORDER TO 0
ELSE
    SELECT 0
    USE (LOCFILE("looking.dbf","DBF","Where is looking?"));
    AGAIN ALIAS looking ;
    ORDER 0
ENDIF
```

```

* *****
*
*      Windows Window definitions
*
* *****
*

```

```

IF NOT WEXIST("_02o1ck9w5")
    DEFINE WINDOW _02o1ck9w5 ;
        AT 0.000, 0.000 ;
        SIZE 20.154, 85.333 ;
        TITLE "Cyprus Hotel" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B" ;
        FLOAT ;
        CLOSE ;
        MINIMIZE ;
        COLOR RGB(,,128,0,0)
    MOVE WINDOW _02o1ck9w5 CENTER
ENDIF

```

```

* *****
*
*      LOOKING/Windows Setup Code - SECTION 2
*
* *****
*

```

#REGION 1

#DEFINE C_DBFEMPTY	'Database is empty, add a record?'
#DEFINE C_EDITS	'Please finish your edits.'
#DEFINE C_TOPFILE	'Top of file.'
#DEFINE C_ENDFILE	'End of file.'
#DEFINE C_BRTITLE	'Locate Record'
#DEFINE C_NOLOCK	'Sorry, could not lock record -- try again later.'
#DEFINE C_ECANCEL	'Edits Canceled.'
#DEFINE C_DELREC	'Delete selected record?'

#DEFINE C_NOFEAT	'Feature not available yet.'
#DEFINE C_NOWIZ	'Wizard application is not available.'
#DEFINE C_MAKEREPO	'Creating report with Report Wizard.'
#DEFINE C_NOREPO	'Could not create report.'
#DEFINE C_DELNOTE	'Deleting records...'
#DEFINE C_READONLY	'Table is read-only. No editing allowed.'
#DEFINE C_NOTABLE	'No table selected. Open table or run query.'
#DEFINE C_BADEXP	'Invalid expression.'
#DEFINE C_LOCWIZ	'Locate WIZARD.APP.'
#DEFINE C_MULTITABLE	'You have multiple related tables. Adding records in not allowed.'

```

MOVE WINDOW '_02o1ck9w5' CENTER
PRIVATE isediting,isadding,wztblarr
PRIVATE wzolddelete,wzolderror,wzoldesc
PRIVATE wzalias, tempcurs,wzlastrec
PRIVATE isreadonly,find_drop,is2table

```

```

IF EMPTY(ALIAS())
    WAIT WINDOW C_NOTABLE
    RETURN
ENDIF

```

```

m.wztblarr= "
m.wzalias=SELECT()
m.isediting=.F.
m.isadding=.F.
m.is2table = .F.
m.wzolddelete=SET('DELETE')
SET DELETED ON
m.tempcurs=SYS(2015) &&used if General field
m.wzlastrec = 1
m.wzolderror=ON('error')
ON ERROR DO wizerrorhandler
wzoldesc=ON('KEY','ESCAPE')
ON KEY LABEL ESCAPE
m.find_drop = IIF(_DOS,0,2)

m.isreadonly=IIF(ISREAD(),.T.,.F.)
IF m.isreadonly

```

```
        WAIT WINDOW C_READONLY TIMEOUT 1
ENDIF
```

```
IF RECCOUNT()=0 AND !m.isreadonly AND fox_alert(C_DBFEMPTY)
    APPEND BLANK
ENDIF
```

```
GOTO TOP
SCATTER MEMVAR MEMO
```

```
* *****
*
*      LOOKING/Windows Screen Layout
*
* *****
*
```

```
#REGION 1
IF WVISIBLE("_02o1ck9w5")
    ACTIVATE WINDOW _02o1ck9w5 SAME
ELSE
    ACTIVATE WINDOW _02o1ck9w5 NOSHOW
ENDIF
```

```
@ 13.615,46.000 GET m.exit_btn ;
    PICTURE "@*HN \<Close" ;
    SIZE 2.077,11.500,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID btn_val('EXIT') ;
    MESSAGE 'Close screen.'
```

```
@ 7.308,22.000 GET aa ;
    PICTURE "@*HN A" ;
    SIZE 1.769,3.500,0.667 ;
    DEFAULT 1 ;
    FONT "MS Sans Serif", 8 ;
    STYLE "B" ;
    VALID _02o1ckafx()
```

```
@ 7.308,27.167 GET wzsstartup ;
```

PICTURE "@*HN B" ;
 SIZE 1.769,3.500,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckaha()
 @ 7.308,32.667 GET wzsshell ;
 PICTURE "@*HN C" ;
 SIZE 1.769,3.500,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckaic()
 @ 7.308,38.000 GET wzsruntime ;
 PICTURE "@*HN D" ;
 SIZE 1.769,3.667,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckaja()
 @ 7.308,43.167 GET wzspreftyp ;
 PICTURE "@*HN E" ;
 SIZE 1.769,3.500,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckakc()
 @ 7.308,48.167 GET wzlshell ;
 PICTURE "@*HN F" ;
 SIZE 1.769,3.333,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckale()
 @ 7.308,52.667 GET wzsshell ;
 PICTURE "@*HN G" ;
 SIZE 1.769,3.667,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;

```

VALID _02o1ckamk()
@ 7.308,57.667 GET wzsdescrip ;
  PICTURE "@*HN H" ;
  SIZE 1.769,3.667,0.667 ;
  DEFAULT 1 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B" ;
  VALID _02o1ckani()
@ 7.308,63.167 GET wzsshell ;
  PICTURE "@*HN I" ;
  SIZE 1.769,3.500,0.667 ;
  DEFAULT 1 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B" ;
  VALID _02o1ckaol()
@ 7.308,68.333 GET wzsruntime ;
  PICTURE "@*HN J" ;
  SIZE 1.769,3.667,0.667 ;
  DEFAULT 1 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B" ;
  VALID _02o1ckapm()
@ 7.385,73.833 GET wzsshell ;
  PICTURE "@*HN K" ;
  SIZE 1.769,3.500,0.667 ;
  DEFAULT 1 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B" ;
  VALID _02o1ckaqu()
@ 7.385,78.833 GET wzatasks ;
  PICTURE "@*HN L" ;
  SIZE 1.769,3.333,0.667 ;
  DEFAULT 1 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B" ;
  VALID _02o1ckary()
@ 10.154,22.000 GET wzsruntime ;
  PICTURE "@*HN M" ;
  SIZE 1.769,3.667,0.667 ;
  DEFAULT 1 ;

```


FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckat6()
 @ 10.154,27.000 GET wzihandle ;
 PICTURE "@*HN N" ;
 SIZE 1.769,3.667,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckaud()
 @ 10.077,32.167 GET wzlshutdow ;
 PICTURE "@*HN O" ;
 SIZE 1.769,3.667,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckavh()
 @ 10.077,37.833 GET wzlmenable ;
 PICTURE "@*HN P" ;
 SIZE 1.769,3.500,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckawp()
 @ 10.077,43.000 GET wzssetexac ;
 PICTURE "@*HN R" ;
 SIZE 1.769,3.667,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckaxy()
 @ 10.077,52.500 GET rborder ;
 PICTURE "@*HN T" ;
 SIZE 1.769,3.500,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckaz6()
 @ 10.077,48.000 GET wzsuses ;
 PICTURE "@*HN S" ;

SIZE 1.769,3.667,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckb0i()
 @ 10.077,57.167 GET wzsuses ;
 PICTURE "@*HN U" ;
 SIZE 1.769,3.500,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckb1u()
 @ 10.077,62.500 GET wzsuses ;
 PICTURE "@*HN V" ;
 SIZE 1.769,3.500,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckb2y()
 @ 10.077,68.167 GET wziprogram ;
 PICTURE "@*HN W" ;
 SIZE 1.769,3.500,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckb46()
 @ 10.077,73.333 GET wziquery ;
 PICTURE "@*HN Y" ;
 SIZE 1.769,3.500,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckb59()
 @ 10.077,78.500 GET wziquery ;
 PICTURE "@*HN Z" ;
 SIZE 1.769,3.500,0.667 ;
 DEFAULT 1 ;
 FONT "MS Sans Serif", 8 ;
 STYLE "B" ;
 VALID _02o1ckb6f()

@ 1.000,27.667 SAY "CYPRUS HOTEL" ;
 FONT "Times New Roman", 18 ;
 STYLE "BT" ;
 COLOR RGB(255,255,255,,,)

@ 6.308,2.500 SAY "SIMPLY" ;
 FONT "MS Sans Serif", 8 ;
 STYLE "BT" ;
 COLOR RGB(255,255,255,,,)

@ 6.308,11.167 SAY "CHOOSE" ;
 FONT "MS Sans Serif", 8 ;
 STYLE "BT" ;
 COLOR RGB(255,255,255,,,)

@ 7.923,5.167 SAY "THE" ;
 FONT "MS Sans Serif", 8 ;
 STYLE "BT" ;
 COLOR RGB(255,255,255,,,)

@ 7.923,10.333 SAY "FIRST" ;
 FONT "MS Sans Serif", 8 ;
 STYLE "BT" ;
 COLOR RGB(255,255,255,,,)

@ 9.615,2.500 SAY "LETTER" ;
 FONT "MS Sans Serif", 8 ;
 STYLE "BT" ;
 COLOR RGB(255,255,255,,,)

@ 9.615,11.333 SAY "OF THE" ;
 FONT "MS Sans Serif", 8 ;
 STYLE "BT" ;
 COLOR RGB(255,255,255,,,)

@ 11.385,5.167 SAY "CUSTOMER" ;
 FONT "MS Sans Serif", 8 ;
 STYLE "BT" ;
 COLOR RGB(255,255,255,,,)

@ 13.231,8.000 SAY "NAME" ;
 FONT "MS Sans Serif", 8 ;
 STYLE "BT" ;
 COLOR RGB(255,255,255,,,)

@ 2.923,27.167 TO 2.923,59.334 ;
 PEN 2, 8 ;
 STYLE "1" ;
 COLOR RGB(255,255,255,255,255,255)

```
@ 5.462,1.000 TO 17.693,83.500 ;  
    PEN 2, 8 ;  
    COLOR RGB(255,255,255,,,) )
```

```
IF NOT WVISIBLE("_02o1ck9w5")  
    ACTIVATE WINDOW _02o1ck9w5  
ENDIF
```

```
* *****  
* *  
* * WindowsREAD contains clauses from SCREEN looking  
* *  
* *****  
*
```

```
READ CYCLE ;  
    ACTIVATE READACT() ;  
    DEACTIVATE READDEAC() ;  
    NOLOCK
```

```
RELEASE WINDOW _02o1ck9w5
```

```
* *****  
* *  
* * Windows Closing Databases  
* *  
* *****  
*
```

```
IF USED("looking")  
    SELECT looking  
    USE  
ENDIF
```

```
SELECT (m.currarea)
```

```
#REGION 0
```



```
SET READBORDER &rborder
```

```
IF m.talkstat = "ON"
```

```
    SET TALK ON
```

```
ENDIF
```

```
IF m.compstat = "ON"
```

```
    SET COMPATIBLE ON
```

```
ENDIF
```

```
* *****
*
*      LOOKING/Windows Cleanup Code
*
* *****
*
```

```
#REGION 1
```

```
SET DELETED &wzolddelete
```

```
SET FIELDS &wzfields
```

```
ON ERROR &wzolderror
```

```
ON KEY LABEL ESCAPE &wzoldesc
```

```
DO CASE
```

```
CASE _DOS AND SET('DISPLAY')='VGA25'
```

```
    @24,0 CLEAR TO 24,79
```

```
CASE _DOS AND SET('DISPLAY')='VGA50'
```

```
    @49,0 CLEAR TO 49,79
```

```
CASE _DOS
```

```
    @24,0 CLEAR TO 24,79
```

```
ENDCASE
```

```
****Procedures****
```

```
* *****
*
*      LOOKING/Windows Supporting Procedures and Functions
*
* *****
*
```

#REGION 1

PROCEDURE readdeac

IF isediting

 ACTIVATE WINDOW '_02o1ck9w5'

 WAIT WINDOW C_EDITS NOWAIT

ENDIF

IF !WVISIBLE(WOUTPUT())

 CLEAR READ

 RETURN .T.

ENDIF

RETURN .F.

PROCEDURE readact

IF !isediting

 SELECT (m.wzalias)

 SHOW GETS

ENDIF

DO REFRESH

RETURN

PROCEDURE wizerrorhandler

 * This very simple error handler is primarily intended

 * to trap for General field OLE errors which may occur

 * during editing from the MODIFY GENERAL window.

 WAIT WINDOW message()

RETURN

PROCEDURE printrec

 PRIVATE sOldError,wizfname,saverec,savearea,tmpcurs,tmpstr

 PRIVATE prnt_btn,p_recs,p_output,pr_out,pr_record

 STORE 1 TO p_recs,p_output

 STORE 0 TO prnt_btn

 STORE RECNO() TO saverec

 m.sOldError=ON('error')

 DO pdialog

 IF m.prnt_btn = 2

 RETURN

 ENDIF

```

IF !FILE(ALIAS()+'FRX')
  m.wizfname=SYS(2004)+'WIZARDS\'+'WIZARD.APP'
  IF !FILE(m.wizfname)
    ON ERROR *
    m.wizfname=LOCFILE('WIZARD.APP','APP',C_LOCWIZ)
    ON ERROR &sOldError
    IF !'WIZARD.APP'$UPPER(m.wizfname)
      WAIT WINDOW C_NOWIZ
      RETURN
    ENDIF
  ENDIF
WAIT WINDOW C_MAKEREPO NOWAIT
  m.savearea=SELECT()
  m.tmpcurs='_'+LEFT(SYS(3),7)
  CREATE CURSOR (m.tmpcurs) (comment m)
  m.tmpstr = '* LAYOUT = COLUMNAR'+CHR(13)+CHR(10)
  INSERT INTO (m.tmpcurs) VALUES(m.tmpstr)
  SELECT (m.savearea)
  DO (m.wizfname) WITH
    ",WZ_QREPO','NOSCRN/CREATE',ALIAS(),m.tmpcurs
    USE IN (m.tmpcurs)
  WAIT CLEAR
  IF !FILE(ALIAS()+'FRX') &&wizard could not create report
    WAIT WINDOW C_NOREPO
    RETURN
  ENDIF
ENDIF

m.pr_out=IIF(m.p_output=1,'TO PRINT NOCONSOLE','PREVIEW')
m.pr_record=IIF(m.p_recs=1,'NEXT 1','ALL')
REPORT FORM (ALIAS()) &pr_out &pr_record
GO m.saverec
RETURN

PROCEDURE BTN_VAL
  PARAMETER m.btnname
  DO CASE
  CASE m.btnname='TOP'
    GO TOP

```

```

        WAIT WINDOW C_TOPFILE NOWAIT
CASE m.btnname='PREV'
    IF !BOF()
        SKIP -1
    ENDIF
    IF BOF()
        WAIT WINDOW C_TOPFILE NOWAIT
        GO TOP
    ENDIF
CASE m.btnname='NEXT'
    IF !EOF()
        SKIP 1
    ENDIF
    IF EOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        GO BOTTOM
    ENDIF
CASE m.btnname='END'
    GO BOTTOM
    WAIT WINDOW C_ENDFILE NOWAIT
CASE m.btnname='LOCATE'
    DO loc_dlog
CASE m.btnname='ADD' AND !isediting &&add record
    isediting=.T.
    isadding=.T.
    =edithand('ADD')
    _curobj=1
    DO refresh
    SHOW GETS
    RETURN
CASE m.btnname='EDIT' AND !isediting &&edit record
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF RLOCK()
        isediting=.T.
        _curobj=1
        DO refresh
        RETURN

```



```

ELSE
    WAIT WINDOW C_NOLOCK
ENDIF
CASE m.btnname='EDIT' AND isediting &&save record
    IF isadding
        =edithand('SAVE')
    ELSE
        GATHER MEMVAR MEMO
    ENDIF
    UNLOCK
    isediting=.F.
    isadding=.F.
    DO refresh
CASE m.btnname='DELETE' AND isediting &&cancel record
    IF isadding
        =edithand('CANCEL')
    ENDIF
    isediting=.F.
    isadding=.F.
    UNLOCK
    WAIT WINDOW C_ECANCEL NOWAIT
    DO refresh
CASE m.btnname='DELETE'
    IF EOF() OR BOF()
        WAIT WINDOW C_ENDFILE NOWAIT
        RETURN
    ENDIF
    IF fox_alert(C_DELREC)
        DELETE
        IF !EOF() AND DELETED()
            SKIP 1
        ENDIF
        IF EOF()
            WAIT WINDOW C_ENDFILE NOWAIT
            GO BOTTOM
        ENDIF
    ENDIF
CASE m.btnname='PRINT'
    DO printrec
    RETURN

```

```

CASE m.btnname='EXIT'
    m.bailout=.T.      &&this is needed if used with FoxApp
    CLEAR READ
    RETURN
ENDCASE
SCATTER MEMVAR MEMO
SHOW GETS
RETURN

```

PROCEDURE REFRESH

```

DO CASE
CASE m.isreadonly AND RECCOUNT()=0
    SHOW GETS DISABLE
    SHOW GET exit_btn ENABLE
CASE m.isreadonly
    SHOW GET add_btn DISABLE
    SHOW GET del_btn DISABLE
    SHOW GET edit_btn DISABLE
CASE (RECCOUNT()=0 OR EOF()) AND !m.isediting
    SHOW GETS DISABLE
    SHOW GET add_btn ENABLE
    SHOW GET exit_btn ENABLE
CASE m.isediting
    SHOW GET find_drop DISABLE
    SHOW GET top_btn DISABLE
    SHOW GET prev_btn DISABLE
    SHOW GET loc_btn DISABLE
    SHOW GET next_btn DISABLE
    SHOW GET end_btn DISABLE
    SHOW GET add_btn DISABLE
    SHOW GET prnt_btn DISABLE
    SHOW GET exit_btn DISABLE
    SHOW GET edit_btn,1 PROMPT "\<Save"
    SHOW GET del_btn,1 PROMPT "\<Cancel"
    ON KEY LABEL ESCAPE DO BTN_VAL WITH 'DELETE'
    RETURN
OTHERWISE
    SHOW GET edit_btn,1 PROMPT "Ed\<it"
    SHOW GET del_btn,1 PROMPT "\<Delete"

```

```

        SHOW GETS ENABLE
    ENDCASE
    IF m.is2table
        SHOW GET add_btn DISABLE
    ENDIF
    ON KEY LABEL ESCAPE
RETURN

```

```

PROCEDURE edithand
    PARAMETER m.paction
    * procedure handles edits
    DO CASE
    CASE m.paction = 'ADD'
        SCATTER MEMVAR MEMO BLANK
    CASE m.paction = 'SAVE'
        INSERT INTO (ALIAS()) FROM MEMVAR
    CASE m.paction = 'CANCEL'
        * nothing here
    ENDCASE
RETURN

```

```

PROCEDURE fox_alert
    PARAMETER wzalrtmess
    PRIVATE alrtbtn
    m.alrtbtn=2
    DEFINE WINDOW _qec1ij2t7 AT 0,0 SIZE 8,50 ;
        FONT "MS Sans Serif",10 STYLE 'B' ;
        FLOAT NOCLOSE NOMINIMIZE DOUBLE TITLE WTITLE()
    MOVE WINDOW _qec1ij2t7 CENTER
    ACTIVATE WINDOW _qec1ij2t7 NOSHOW
    @ 2,(50-txtwidth(wzalrtmess))/2 SAY wzalrtmess;
        FONT "MS Sans Serif", 10 STYLE "B"
    @ 6,18 GET m.alrtbtn ;
        PICTURE "@*HT \<OK; \? \! \<Cancel" ;
        SIZE 1.769,8.667,1.333 ;
        FONT "MS Sans Serif", 8 STYLE "B"
    ACTIVATE WINDOW _qec1ij2t7
    READ CYCLE MODAL
    RELEASE WINDOW _qec1ij2t7

```

RETURN m.alrtbtn=1

PROCEDURE pdialog

```
    DEFINE WINDOW _qjn12zbvh ;
        AT 0.000,0.000 ;
        SIZE 13.231,54.800 ;
        TITLE "Microsoft FoxPro" ;
        FONT "MS Sans Serif", 8 ;
        FLOAT NOCLOSE MINIMIZE SYSTEM
    MOVE WINDOW _qjn12zbvh CENTER
    ACTIVATE WINDOW _qjn12zbvh NOSHOW
    @ 2.846,33.600 SAY "Output:" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
    @ 2.846,4.800 SAY "Print:" ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
    @ 4.692,7.200 GET m.p_recs ;
        PICTURE "@*RVN \<Current Record;\<All Records" ;
        SIZE 1.308,18.500,0.308 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
    @ 4.692,36.000 GET m.p_output ;
        PICTURE "@*RVN \<Printer;Pre\<view" ;
        SIZE 1.308,12.000,0.308 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "BT"
    @ 10.154,16.600 GET m.prnt_btn ;
        PICTURE "@*HT P\<rint;Ca\<ncel" ;
        SIZE 1.769,8.667,0.667 ;
        DEFAULT 1 ;
        FONT "MS Sans Serif", 8 ;
        STYLE "B"
    ACTIVATE WINDOW _qjn12zbvh
    READ CYCLE MODAL
    RELEASE WINDOW _qjn12zbvh
RETURN
```



```

PROCEDURE loc_dlog
  PRIVATE gfields,i
  DEFINE WINDOW wzlocate FROM 1,1 TO 20,40;
  SYSTEM GROW CLOSE ZOOM FLOAT FONT "MS Sans
Serif",8
  MOVE WINDOW wzlocate CENTER
  m.gfields=SET('FIELDS',2)
  IF !EMPTY(RELATION(1))
    SET FIELDS ON
    IF m.gfields # 'GLOBAL'
      SET FIELDS GLOBAL
    ENDIF
    IF EMPTY(FLDLIST())
      m.i=1
      DO WHILE !EMPTY(OBJVAR(m.i))
        IF ATC('M.',OBJVAR(m.i))=0
          SET FIELDS TO (OBJVAR(m.i))
        ENDIF
        m.i = m.i + 1
      ENDDO
    ENDIF
  ENDIF
  BROWSE WINDOW wzlocate NOEDIT NODELETE ;
  NOMENU TITLE C_BRTITLE
  SET FIELDS &gfields
  SET FIELDS OFF
  RELEASE WINDOW wzlocate
RETURN

```

```

* *****
*
* * _02O1CKAFX      aa VALID
*
* * Function Origin:
*
* * From Platform:   Windows

```

```

*   * From Screen:      LOOKING,   Record Number:   4
*   * Variable:         aa
*   * Called By:        VALID Clause
*   * Object Type:      Push Button
*   * Snippet Number:   1
FUNCTION _02o1ckafx   && aa VALID
#REGION 1
DO C:/TAISIR/A.QPR
*   * _02O1CKAHA       wzsstartup VALID
*   *
*   * Function Origin:
*   *
*   * From Platform:    Windows
*   * From Screen:      LOOKING,   Record Number:   5
*   * Variable:         wzsstartup
*   * Called By:        VALID Clause
*   * Object Type:      Push Button
*   * Snippet Number:   2
*   *
*   *****
*
FUNCTION _02o1ckaha   && wzsstartup VALID
#REGION 1
DO C:/TAISIR/B.QPR

*   *****
*   *
*   * _02O1CKAIC       wzsshell VALID
*   *
*   * Function Origin:
*   *
*   * From Platform:    Windows
*   * From Screen:      LOOKING,   Record Number:   6
*   * Variable:         wzsshell
*   * Called By:        VALID Clause
*   * Object Type:      Push Button
*   * Snippet Number:   3
*   *
*   *****
*

```

```
FUNCTION _02o1ckaic  && wzsshell VALID
#REGION 1
DO C:/TAISIR/C.QPR
```

```
* *****
*
* * _02O1CKAJA      wzsruntime VALID
*
* * Function Origin:
*
* * From Platform:   Windows
* * From Screen:    LOOKING,   Record Number:  7
* * Variable:       wzsruntime
* * Called By:      VALID Clause
* * Object Type:    Push Button
* * Snippet Number:  4
*
* *****
*
```

```
FUNCTION _02o1ckaja  && wzsruntime VALID
#REGION 1
DO C:/TAISIR/D.QPR
```

```
* *****
*
* * _02O1CKAKC      wzspreftyp VALID
*
* * Function Origin:
*
* * From Platform:   Windows
* * From Screen:    LOOKING,   Record Number:  8
* * Variable:       wzspreftyp
* * Called By:      VALID Clause
* * Object Type:    Push Button
* * Snippet Number:  5
*
* *****
*
```

```
FUNCTION _02o1ckakc  && wzspreftyp VALID
#REGION 1
```

DO C:/TAISIR/E.QPR

```
* *****
*
* * _02O1CKALE      wzlshell VALID
*
* * Function Origin:
*
* * From Platform:   Windows
* * From Screen:     LOOKING,   Record Number:  9
* * Variable:        wzlshell
* * Called By:       VALID Clause
* * Object Type:     Push Button
* * Snippet Number:  6
*
* *****
*
```

FUNCTION _02o1ckale && wzlshell VALID

#REGION 1

DO C:/TAISIR/F.QPR

```
* *****
*
* * _02O1CKAMK      wzsshell VALID
*
* * Function Origin:
*
* * From Platform:   Windows
* * From Screen:     LOOKING,   Record Number: 10
* * Variable:        wzsshell
* * Called By:       VALID Clause
* * Object Type:     Push Button
* * Snippet Number:  7
*
* *****
*
```

FUNCTION _02o1ckamk && wzsshell VALID

#REGION 1

DO C:/TAISIR.QPR


```

* *****
*
* * _02O1CKANI      wzsdDescrip VALID
*
* * Function Origin:
*
* * From Platform:   Windows
* * From Screen:     LOOKING,   Record Number:  11
* * Variable:        wzsdDescrip
* * Called By:       VALID Clause
* * Object Type:     Push Button
* * Snippet Number:   8
*
* *****

```

```

FUNCTION _02o1ckani  && wzsdDescrip VALID
#REGION 1
DO C:/TAISIR/H.QPR

```

```

* *****
*
* * _02O1CKAOL      wzsshell VALID
*
* * Function Origin:
*
* * From Platform:   Windows
* * From Screen:     LOOKING,   Record Number:  12
* * Variable:        wzsshell
* * Called By:       VALID Clause
* * Object Type:     Push Button
* * Snippet Number:   9
*
* *****

```

```

FUNCTION _02o1ckaol  && wzsshell VALID
#REGION 1
DO C:/TAISIR/I.QPR

```

```

* *****
*

```

```

*   * _02O1CKAPM      wzsruntime VALID
*   *
*   * Function Origin:
*   *
*   * From Platform:    Windows
*   * From Screen:      LOOKING,   Record Number: 13
*   * Variable:         wzsruntime
*   * Called By:        VALID Clause
*   * Object Type:      Push Button
*   * Snippet Number:   10
*   *
*   *****
*

```

```

FUNCTION _02o1ckapm  && wzsruntime VALID
#REGION 1
DO C:/TAISIR/J.QPR

```

```

*   *****
*   *
*   * _02O1CKAQU      wzsshell VALID
*   *
*   * Function Origin:
*   *
*   * From Platform:    Windows
*   * From Screen:      LOOKING,   Record Number: 14
*   * Variable:         wzsshell
*   * Called By:        VALID Clause
*   * Object Type:      Push Button
*   * Snippet Number:   11
*   *
*   *****
*

```

```

FUNCTION _02o1ckaqu  && wzsshell VALID
#REGION 1
DO C:/TAISIR/K.QPR

```

```

*   *****
*   *
*   * _02O1CKARY      wzatasks VALID
*   *

```

```

*   * Function Origin:
*   *
*   * From Platform:   Windows
*   * From Screen:     LOOKING,   Record Number: 15
*   * Variable:        wzatasks
*   * Called By:       VALID Clause
*   * Object Type:     Push Button
*   * Snippet Number:  12
*   *
*   *****

```

```

FUNCTION _02o1ckary  && wzatasks VALID
#REGION 1
DO C:/TAISIR/L.QPR

```

```

*   *****
*   *
*   * _02O1CKAT6      wzsruntime VALID
*   *
*   * Function Origin:
*   *
*   * From Platform:   Windows
*   * From Screen:     LOOKING,   Record Number: 16
*   * Variable:        wzsruntime
*   * Called By:       VALID Clause
*   * Object Type:     Push Button
*   * Snippet Number:  13
*   *
*   *****

```

```

FUNCTION _02o1ckat6  && wzsruntime VALID
#REGION 1
DO C:/TAISIR/M.QPR

```

```

*   *****
*   *
*   * _02O1CKAUD      wzihandle VALID
*   *
*   * Function Origin:
*   *

```

```

*      * From Platform:    Windows
*      * From Screen:     LOOKING,   Record Number: 17
*      * Variable:        wzihandle
*      * Called By:       VALID Clause
*      * Object Type:     Push Button
*      * Snippet Number:  14
*      *
*      *****
*

```

```

FUNCTION _02o1ckaud    && wzihandle VALID
#REGION 1
DO C:/TAISIR/N.QPR

```

```

*      *****
*      *
*      * _02O1CKAVH      wzlshutdow VALID
*      *
*      * Function Origin:
*      *
*      * From Platform:    Windows
*      * From Screen:     LOOKING,   Record Number: 18
*      * Variable:        wzlshutdow
*      * Called By:       VALID Clause
*      * Object Type:     Push Button
*      * Snippet Number:  15
*      *
*      *****
*

```

```

FUNCTION _02o1ckavh    && wzlshutdow VALID
#REGION 1
DO C:/TAISIR/O.QPR

```

```

*      *****
*      *
*      * _02O1CKAWP      wzlmenable VALID
*      *
*      * Function Origin:
*      *
*      * From Platform:    Windows
*      * From Screen:     LOOKING,   Record Number: 19

```



```

* * Variable:      wzlmenable
* * Called By:     VALID Clause
* * Object Type:   Push Button
* * Snippet Number: 16
FUNCTION _02o1ckawp  && wzlmenable VALID
#REGION 1
DO C:/TAISIR/P.QPR
* * _02O1CKAXY      wzssetexac VALID
* *
* * Function Origin:
* * From Platform:  Windows
* * From Screen:    LOOKING,   Record Number: 20
* * Variable:       wzssetexac
* * Called By:      VALID Clause
* * Object Type:    Push Button
* * Snippet Number: 17
FUNCTION _02o1ckaxy  && wzssetexac VALID
#REGION 1
DO C:/TAISIR/R.QPR

* *****
* *
* * _02O1CKAZ6      rborder VALID
* *
* * Function Origin:
* *
* * From Platform:  Windows
* * From Screen:    LOOKING,   Record Number: 21
* * Variable:       rborder
* * Called By:      VALID Clause
* * Object Type:    Push Button
* * Snippet Number: 18
FUNCTION _02o1ckaz6  && rborder VALID
#REGION 1
DO C:/TAISIR/R.QPR
* * _02O1CKB0I      wzsuses VALID
* *
* * Function Origin:
* * From Platform:  Windows
* * From Screen:    LOOKING,   Record Number: 22

```

```

*   * Variable:      wzsuses
*   * Called By:     VALID Clause
*   * Object Type:   Push Button
*   * Snippet Number: 19
FUNCTION _02o1ckb0i  && wzsuses VALID
#REGION 1
DO C:/TAISIR/S.QPR
*   * _02O1CKB1U      wzsuses VALID
*   * Function Origin:
*   * From Platform:  Windows
*   * From Screen:    LOOKING,   Record Number: 23
*   * Variable:      wzsuses
*   * Called By:     VALID Clause
*   * Object Type:   Push Button
*   * Snippet Number: 20
FUNCTION _02o1ckb1u  && wzsuses VALID
#REGION 1
DO C:/TAISIR/U.QPR
*   * _02O1CKB2Y      wzsuses VALID
*   * Function Origin:
*   *
*   * From Platform:  Windows
*   * From Screen:    LOOKING,   Record Number: 24
*   * Variable:      wzsuses
*   * Called By:     VALID Clause
*   * Object Type:   Push Button
*   * Snippet Number: 21
FUNCTION _02o1ckb2y  && wzsuses VALID
#REGION 1
DO C:/TAISIR/V.QPR
*   * _02O1CKB46      wziprogram VALID
*   *
*   * Function Origin:
*   * From Platform:  Windows
*   * From Screen:    LOOKING,   Record Number: 25
*   * Variable:      wziprogram
*   * Called By:     VALID Clause
*   * Object Type:   Push Button
*   * Snippet Number: 22
FUNCTION _02o1ckb46  && wziprogram VALID

```

#REGION 1

DO C:/TAISIR/W.QPR

* * _02O1CKB59 wziquery VALID
* * Function Origin:
* * From Platform: Windows
* * From Screen: LOOKING, Record Number: 26
* * Variable: wziquery
* * Called By: VALID Clause
* * Object Type: Push Button
* * Snippet Number: 23

FUNCTION _02o1ckb59 && wziquery VALID

#REGION 1

DO C:/TAISIR/Y.QPR

* * _02O1CKB6F wziquery VALID
* * Function Origin:
* * From Platform: Windows
* * From Screen: LOOKING, Record Number: 27
* * Variable: wziquery
* * Called By: VALID Clause
* * Object Type: Push Button
* * Snippet Number: 24

FUNCTION _02o1ckb6f && wziquery VALID

#REGION 1

DO C:/TAISIR/Z.QPR

CONCLUSION

To adopt the technology at Tourism Business, we have to have professional Tourism management System.

At the beginning we have to mention that to have management system in hotel provide us such as advantages these are;

- . It will provide the efficiency of production
- . It will improve the quality of services
- . It will lead to the generation of new services (image communication, satellite communication, teleconferencing etc.)

So that have exact management programs of hotel, First of all we have to get each details of the particular establishment. (Number of rooms, variety of rooms, agreement of hotels, agreements of agencies etc.) So that we arrange the management program according to the particular hotel.

Consequently, once we made a excellent hotel management program, after that according to the radical changes in the hotel business, we have to update the program. Because, no player in this industry will not escape the impacts of the information technology