NEAR EAST UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

COM 400

GRADUATION PROJECT

LIBRARY AND BOOKSTORE AND DATABASE

SUBMITTED TO

MRS BESIME ERIN

SUBMITTED BY

IMRAN SOHAIL & BEDRIYE OZLEM PEKMEZ 960694 950204

Tables of Contents

1-Switch Board

•Switch Board for Departments and Currency

- Switch Board for Reports
 - Report of Book Issue
 - Report of Book Sale

2-Forms

Department Form •Codes for Form Department

Currency Form

•Codes for Form Currency

Book Form

•Codes for Form Books

Students Form

•Codes for Form Student

Book Issue Form

•Codes for Form Issue Book

• Codes for Form Issue Book II

Book Sale Form

• Codes for Form Book Sale

• Codes for Form Book Sale II Check Fine Form

• Codes For Form Check Fine

3- Conclusion

Acknowledgement:

First of all we are thankful to our God who is most kind and merciful. We are thankful to our parents who enabled us to do our graduation and do what we need, They did all those things we want. And at the and we want to say thanks to our friends who helped us in completion of our project. In the friends the most upper names are MUHAMMAD ATIF Malik, MUHAMMAD NAEEM,ILKER YIGIT, MEHET BOZKAN,IHSAN ZENGIN AND NECLA YIĞIT. We are also thankful to all the teachers who brought us at this point.

Introduction

Welcome to *Microsoft Access 2000*, In our Project you'll find out how to build applications that you and others can use to effectively manage data. In addition, you'll learn how to use Visual Basic® for Applications—the programming language included in Microsoft Access—to go beyond what you can accomplish by using only the Microsoft Access interface. With Visual Basic, you can extend and customize your application to fit your needs and those of your users.

If you're new to Microsoft Access, look first at Getting Results with Microsoft Access 2000 for information on creating the objects that make up a Microsoft Access database. In addition, consult the Office Assistant with questions that arise as you learn Microsoft Access.

Our Project has Four Sections:

The first section shows you how to tie database objects together into a coherent system and introduces Visual Basic for Applications, the programming language used in Microsoft Access. Examples in this section are based on the Orders sample application included with the Microsoft Access sample applications.

The second section, describes the elements of Visual Basic in more detail, including an overview of the objects and collections you can work within Visual Basic and the events you can respond to. In addition, it shows you how to use the Visual Basic debugging tools to test and debug your Visual Basic code, and explains how to respond to errors that occur while your code is running. The third section, shows you how to accomplish complex data management tasks in your application, including manipulating sets of records in code, developing an application for multiple users, communicating with other applications such as Microsoft Excel or Microsoft Word, and using library databases and dynamic-link libraries. In addition, the third section explains how to optimize, secure, and deliver your application to users.

The fourth section, which includes, explores powerful features that address special development needs, such as using ActiveX[™] controls, creating your own wizards and add-ins, working with external or client/server data, implementing database replication, and developing applications for the Internet and the World Wide Web.

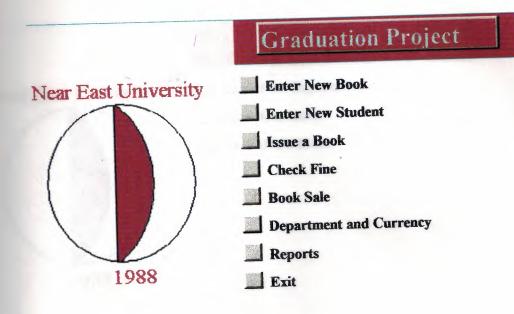
What's New in Microsoft Access 2000

Microsoft Access for Windows® 98, Windows NT and windows 2000 introduced many new features for application development. Microsoft Access 2000 introduces additional new features, as well as enhancements to existing features. The following sections highlight new features introduced in both versions.

Note If you have applications that were created with Microsoft Access 97 or earlier, you'll need to consider several important changes that affect your existing code and database objects. For more information, search the Help index for "converting databases."

Visual Basic for Applications Programming Language

Visual Basic for Applications replaces Access Basic, which was used in versions 2.0 and earlier, as the programming language for Microsoft Access. This is the same language you use to program in Microsoft Word, Microsoft Excel, and Microsoft Project, as well as the same language found in Microsoft Visual Basic.



Switch Board:

The first form, which we can see while opening the project, is the Switchboard. It contains all the buttons using them we can enter into the project and can work accordingly.

The main Items of switchboard are shown in the figure. It

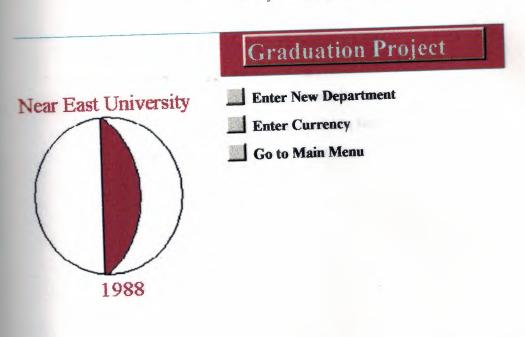
contains

- 1. Enter New Book
- 2. Enter New Student
- 3. Issue a Book
- 4. Check Fine
- 5. Book Sale
- 6. Department And Currency
- 7. Reports
- 8. Exit

Before going in detail of these all buttons, I want to tell you that in one switchboard we can add maximum 8 buttons, if we have more than 8 functions to do, So for this purpose we can add more than one switchboard, as we did here,

Caption "Department and Currency" is the switch to next switchboard, which consists of three fields

3



- 1. Enter New Department
- 2. Enter New Currency
- 3. Go to Main Menu

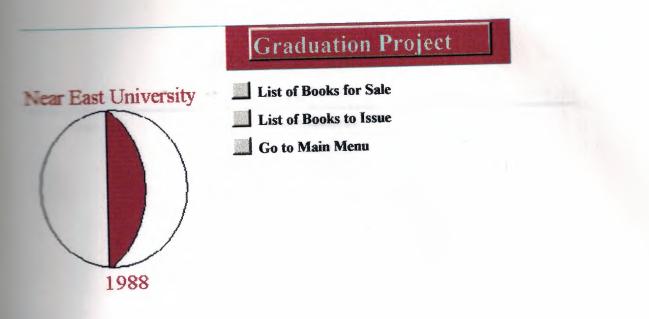
"Go to Main Menu" is the way back to the main menu or the first switchboard. The first button is to open the form, which is using to add new departments. And the same for the second, it will open the form, which is for the addition of new currency. We can see the detail of these forms later in the detail of forms.

If we have a look right down of this button there is "*Reports*" button which will guide us to the switch board of reports on which we have the buttons of reports clicking them we can see the reports.

Now if we fix a gaze on the other buttons of main switchboard, the first buttons we can see is the *"Enter New Book"*. This button leads us to the form for entering new books and the same for the others that every buttons is opening its related form or switchboard.

The last button is "*Exit*" to exit from the application or to end the program.

4



Switch Board for Reports.

Books For Sale

Author

Book Title

Hamlet

William Shakespeare

Quantity 20

Title Author	ISBN	Department
	Days (bern	
*		
in the second		
nin a sana sa mangana ang kangana da sana kang kang kang kang kang kang kang k		a line of the

Library and Book Store Database

ave
and a second sec

Form Department

		400 y 200 J 444 A 7744 4 1 104	eire'
1. 1. 1. 1. 1.			
and the second	<u>.</u>		
Ex	if		
	E	Exit	Exit

Form Currency

Forms:

The form is a screen view on which we can edit or add our new data entry. In our Project we have nine different forms, Names are listed in the switchboard. As we have described in the section of switchboard. Now we are going to give a detail of all forms individually.

We will describe it in the routine of usage. The first form we will use to add data of our need that is the department and currency forms. We will add some departments and currency using this form.

Department Form

In this form we have two fields and three command buttons. The first field is the department number, which is describing the serial number of departments and the second field is for the departmental names. The first command button is the "*New*" button, which is for adding a new department. When you will press this button a new number will appear in the department number field and you can write new name in the department name field. After doing this you have to press the second button, which is "*Save*" for saving the data you entered. And the third button, as its caption is showing that it is "*Exit*" button, is to exit from the form.

Codes for Form Department

Option Compare Database Option Explicit

Private Sub cmdsave_Click() Dim rssave As New ADODB.Recordset

rssave.Open "select * from tbldept where flddeptno = "' + Str(flddeptno) + "", CurrentProject.Connection, adOpenDynamic, adLockPessimistic, adCmdText

> rssave.AddNew rssave("flddeptno") = flddeptno rssave("flddeptname") = flddeptname

rssave.Update

rssave.Close

End Sub

Private Sub Command5_Click() On Error GoTo Err_Command5_Click

DoCmd.Close

Exit_Command5_Click: Exit Sub

Err_Command5_Click: MsgBox Err.Description Resume Exit_Command5_Click

End Sub

Private Sub Cmdnew_Click() Dim rs As New ADODB.Recordset

rs.Open "select max(cint(flddeptno))+1,count(*) from tbldept", CurrentProject.Connection, adOpenDynamic, adLockPessimistic, adCmdText

If rs(1) = 0 Then flddeptno = 1 Else flddeptno = rs(0) End If flddeptname = ""

rs.Close End Sub

Codes for Form Currency

```
Option Compare Database
```

```
Private Sub Cmdnew_Click()
```

```
Dim rsnew As New ADODB.Recordset
```

```
rsnew.Open "select max(cint(fldcurid))+1,count(*) from tblcur",
CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic,
adCmdText
```

```
If rsnew(1) = 0 Then
fldcurid = 1
Else
'MsgBox rsnew(0)
fldcurid = rsnew(0)
rsnew.Close
End If
fldcurname = ""
```

End Sub

```
Private Sub cmdsave_Click()
Dim rssave As New ADODB.Recordset
rssave.Open "select * from tblcur ", CurrentProject.Connection,
adOpenForwardOnly, adLockPessimistic, adCmdText
rssave.AddNew
rssave("fldcurid") = fldcurid
rssave("fldcurname") = fldcurname
rssave.Update
rssave.Close
End Sub
```

Private Sub Command6_Click() On Error GoTo Err_Command6_Click

DoCmd.Close

Exit_Command6_Click: Exit Sub Err_Command6_Click: MsgBox Err.Description Resume Exit_Command6_Click

End Sub

Currency Form

The currency form is working same as the department is, here also we have two fields and three command, the first field is for rency number to give the serial number of currency and the second is for rency name. And the rest of three command buttons are working as in *Department Form*.

Book Title	
Author ISBN	
1351	
Save	New Exit
	Book Form I
	Book Form I
Department	Book Form I
Department Language	Book Form I
	Book Form I
Language	

Book Form II

Book Form

As the name of the form is showing the characteristics of the form that this form is concerned with the books of library. So in this form have two parts name "Book Name" and "Book Detail". In Book Name have four main fields and three buttons. The first field, which has no label, is giving the number to count that how many books we have entered. The second field is "Book Title" which is for the Book Title. The third is for the Author Name and the last on is for the ISBN of that Book. The three command Buttons or working as the caption or showing the properties of them, The first one is New button whenever we want to add a book we have to press this button. The second one is for Saving the record, which we have entered but before pressing the Save button we have to complete the whole form including information of Book Detail.

In this Book Detail Form we have actually six fields but in picture we can see only three. The first field is a Combo Box, which have the list of all departments, which we have entered in the "Department Form". The purpose of this combo box is to give the identification to book that this book is of which department. The second field is the language field to show that the book is in which language.

The purpose of third check box, which is labeled, as "For Sale" is to show that whether this book is for sale or for issue, When you will check this button three more fields will appear Labeled as "Unit Price" "Currency" and "Quantity" respectively. In first text box we have to enter the price of that book, the second one, which is actually a combo box, is showing all the currencies, which we entered in "<u>Currency Form</u>". Here we will select that the price of book is in which currency. And the last one, which is "Quantity", it will show that what is the quantity of that specific book we have in our library for sale.

If we are checking this box the program will consider that this book is for issuing not for sale. After completing this information of the books we will go back to the "Book Name" side and we will press Save Button.

Codes for Form Books

Option Compare Database Option Explicit

Private Sub chkforsale_Click() If chkforsale = True Then txtqty.Visible = True txtprice.Visible = True txtprice.Enabled = True cmbcur.Visible = True Else txtqty.Visible = False txtprice.Visible = False txtprice.Enabled = False cmbcur.Visible = False End If End Sub

Private Sub Command10_Click() On Error GoTo Err Command10_Click

DoCmd.GoToRecord,, acFirst

Exit_Command10_Click: Exit Sub

Err Command10_Click:

MsgBox Err.Description Resume Exit_Command10_Click

End Sub

Private Sub Command11_Click() On Error GoTo Err Command11_Click

DoCmd.GoToRecord,, acPrevious

Exit_Command11_Click: Exit Sub

Err_Command11_Click: MsgBox Err.Description Resume Exit_Command11_Click

End Sub

Private Sub Command12_Click() On Error GoTo Err_Command12_Click

DoCmd.GoToRecord,, acNext

Exit_Command12_Click: Exit Sub

Err_Command12_Click: MsgBox Err.Description Resume Exit_Command12_Click

End Sub

Command13_Click() Command13_Click

DoCmd.GoToRecord,, acLast

Exit_Command13_Click: Exit Sub Err_Command13_Click: MsgBox Err.Description Resume Exit_Command13_Click

End Sub

```
Private Sub Command14_Click()
Dim rsnew As New ADODB.Recordset
rsnew.Open "select max(cint(fldbookid))+1,count(*) from
blbookmain", CurrentProject.Connection, adOpenForwardOnly,
adLockPessimistic, adCmdText
```

If rsnew(1) = 0 Then fldbookid = 1 Else fldbookid = rsnew(0) rsnew.Close End If chkforsale = False fldbookname = "" fldauthor = "" fldauthor = "" fldlanguage = "" txtprice = "" txtprice = False txtprice.Enabled = False cmbcur.Visible = False

End Sub

Private Sub Command15_Click() On Error GoTo Err_Command15_Click

DoCmd.Close

Exit_Command15_Click: Exit Sub

Err_Command15_Click: MsgBox Err.Description Resume Exit_Command15_Click

End Sub

Private Sub Command18 Click() Dim rssave As New ADODB.Recordset rssave.Open "select * from tblbookmain ", CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic, adCmdText rssave.AddNew rssave("fldbookid") = fldbookid rssave("fldbookname") = fldbookname rssave("fldauthor") = fldauthor rssave("fldisbn") = fldisbn rssave("flddept") = flddept rssave("fldlanguage") = fldlanguage rssave("fldprice") = txtprice rssave("fldforsale") = chkforsale rssave("fldcur") = cmbcur rssave("fldqty") = txtqty rssave.Update rssave.Close

End Sub

Private Sub Form_Load() chkforsale = False txtqty.Visible = False

HOLE ROOM \$1

End Sub

Students Form:

This form is to add the information of Students, that means that when a student is going to issue a book that has to register in the

	and the second second second
Student ID	950204
First Name	B.Pekmez
Last Name	Ozlem
Department	Computer Engineering
14 4 >	Ne D+

Student Form I

Address	Gazantepe
Tel. Phone	2258745
E- Mail	bedisp@hotmail.com

Student Form II

In this form we have seven fields labeled as "Student ID" First Name" "Last Name" "Department" "Address" "Telephone Number" E-Mail". The first field is the student ID, before entering any new student's monation we have to press new button, after this we will enter the ID of Student. As we will lose the focus from this field we will get a message from program, actually there are two types of messages. If the student ID is entering first time the program will give a message.

The program could not find the ID, Do you want to enter this ID"

There are two options Yes and No. If you will choose yes then the program will set the focus on the field "First Name". And you can continue with entering the information. And you will choose the second option that is NO the program will erase student ID to enter a new ID.

If you forget to enter a ID and you try to enter directly the name the program will give you and error message which is

You can not enter a name without entering ID"

This message will appear and on clicking ok the program will erase the name and will set the focus on ID. The rest of the fields are doing the normal operations as labeled on the form.

We have some extra buttons with some images on them they are for moving the data, the most left side button is for the "First Record" the next is for the Previous Record, The next to this button is for the "Next Record" and the fourth one is for "Last Record". The most right side button is the "Exit" button to close the form.

Codes for Form Student

Option Compare Database Option Explicit Dim a

Private Sub Command10_Click() On Error GoTo Err Command10 Click

a = 1 DoCmd.GoToRecord , , acNewRec

Exit_Command10_Click: Exit Sub

Err_Command10_Click: MsgBox Err.Description Resume Exit_Command10_Click

End Sub

Private Sub Command11_Click() On Error GoTo Err_Command11_Click

DoCmd.GoToRecord,, acFirst

Exit_Command11_Click: Exit Sub

Err Command11 Click:

MsgBox Err.Description Resume Exit_Command11_Click

End Sub

Private Sub Command12_Click() On Error GoTo Err_Command12_Click

DoCmd.GoToRecord,, acPrevious

Exit_Command12_Click: Exit Sub

Err_Command12_Click: MsgBox Err.Description Resume Exit_Command12_Click

End Sub

Private Sub Command13_Click() On Error GoTo Err_Command13_Click

DoCmd.GoToRecord,, acNext

Exit_Command13_Click: Exit Sub

Err_Command13_Click: MsgBox Err.Description Resume Exit_Command13_Click

End Sub

Command14_Click() Command14_Click() Command14_Click

DoCmd.GoToRecord,, acLast

Exit_Command14_Click: Exit Sub

Err_Command14_Click: MsgBox Err.Description Resume Exit_Command14_Click End Sub Private Sub Command17_Click() On Error GoTo Err Command17 Click

DoCmd.Close

Exit_Command17_Click: Exit Sub

Err_Command17_Click: MsgBox Err.Description Resume Exit_Command17_Click

End Sub

Private Sub Form_Load() a = 0 End Sub

Private Sub txtstdfname LostFocus()

If IsNull(txtstudentid) Or txtstudentid = "" Then

MsgBox "You are not allowed to Enter a Student without his/her ID", vbCritical, "Graduation Project" txtstdfname = ""

txtstudentid.SetFocus End If End Sub

```
Dim rs As New ADODB.Recordset
Dim rsstd As New ADODB.Recordset
Dim choice
```

```
rs.Open "select fldstdfname,fldstdlname from tblstdmain where

dstudentid = "" + txtstudentid + """, CurrentProject.Connection,

dOpenForwardOnly, adLockPessimistic, adCmdText

rsstd.Open "select fldstudentid from tblstdmain",

CurrentProject.Connection, adOpenDynamic, adLockPessimistic,
```

adCmdText

If txtstudentid = rsstd(0) And a = 1 Then MsgBox "The student Number " + txtstudentid + " Already Exist ", vbExclamation, " Graduation Project" txtstdfname = "" txtstdfname = "" txtstudentid = "" txtstudentid = ""

```
Else
If rs.EOF = True Then
If Len(txtstudentid) = 6 Then
```

```
choice = MsgBox("The Program could not find: " +
txtstudentid + " Do You Want to Enter this ID", vbYesNo, "Graduation
Project")
```

If choice = 6 Then ' DoCmd.OpenForm "frmstdmain", acNormal

'txtstudentid = ""

'txtstudentid.SetFocus txtstdfname = "" txtstdlname = ""

txtstdfname.SetFocus Else txtstudentid = "" txtstdfname.SetFocus txtstdlname = "" txtstudentid.SetFocus txtstdfname = ""

End If

Else If txtstudentid = "" Then Exit Sub

Else

MsgBox txtstudentid + " is not a valid Student ID",

```
"Critical, "Graduation Project"
```

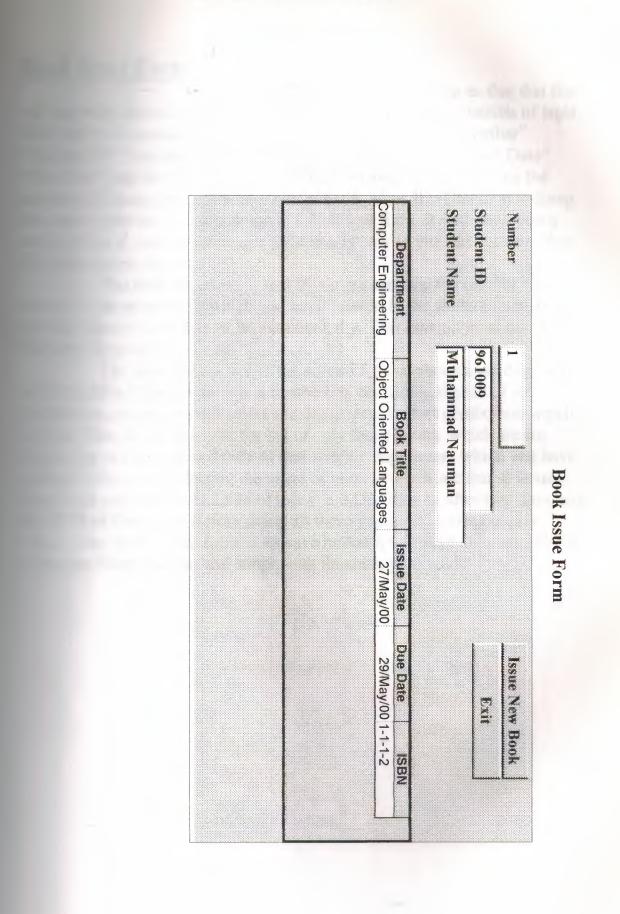
txtstudentid = "" txtstdfname.SetFocus txtstudentid.SetFocus txtstdfname = "" txtstdlname = "" End If End If Else

```
txtstdfname = rs(0)
txtstdlname = rs(1)
txtstudentid = ""
'txtstdfname.SetFocus
'txtstudentid.SetFocus
```

End If End If

rs.Close

End Sub



Book Issue Form:

As the name of the form is showing us that this fine will use when some one is going to issue a book. This form consists of eight fields and two command buttons. The fields are labeled as "Number" Student ID" "Student Name" "Department" "Book Title" "Issue Date" Due Date" and the "ISBN". The first field Number is just showing the same operation as it was doing in the student ID. This field is doing the same operation as it was doing in the student form. It will show you a message box if you are entering a ID which is new or you did not enter that ID in the student form.

The difference here is that if you are entering the number which is already in the program or which you have entered in the student form, it will give the name of that student ID automatically. You have no need to enter the name by yourself.

The next thing which is the second form is showing as datasheet, its first field is Department, it is a combo box containing the list of all departments, when you will select the department in next combo box which is Book Title. It will give you the list of only those books, which are for Issuing but not for sale and only of that specific department which you have selected before. On Selecting the name of book you will see that in Issue Date Field you will see the date of today and Due date is After two days and the ISBN of that book. Before doing all these you have to press button "Issue New Book". And there is no save button it will save automatically as in Student Form. So you will simply exit finishing your work. Codes for Form Issue Book

Option Compare Database

Private Sub Cmdnew_Click() Dim rsnew As New ADODB.Recordset

DoCmd.GoToRecord , , acNewRec

rsnew.Open "select max(cint(fldissueno))+1,count(*) from bissuedet", CurrentProject.Connection, adOpenForwardOnly, bissuedet, adCmdText

> If rsnew(1) = 0 Then fldissueno = 1 Else

fldissueno = rsnew(0) rsnew.Close txtstudentid = "" txtstudentname = ""

End If

End Sub

Private Sub cmdnewstd_Click() DoCmd.OpenForm "frmstdmain", acNormal End Sub Private Sub cmdsave_Click()

Dem rssave As New ADODB.Recordset

rssave.Open "select * from tblissuedet", CurrentProject.Connection,

rssave("fldstudentid") = txtstudentid
rssave("fldstudentname") = txtstudentname
rssave.Update

rssave.Close

End Sub

Private Sub Form_Load() DoCmd.Maximize End Sub

```
Private Sub txtstudentid_AfterUpdate()
Dim rs As New ADODB.Recordset
Dim choice
    rs.Open "select fldstdfname,fldstdlname from tblstdmain where
distudentid = "" + txtstudentid + """, CurrentProject.Connection,
dOpenForwardOnly, adLockPessimistic, adCmdText
    If rs.EOF = True Then
        If Len(txtstudentid) = 6 Then
        choice = MsgBox("The Program could not find: " +
txtstudentid + " Do You Want to Enter this ID", vbYesNo, "Graduation
Project")
    If choice = 6 Then
```

DoCmd.OpenForm "frmstdmain", acNormal

txtstudentid = "" txtstudentname.SetFocus txtstudentid.SetFocus txtstudentname = "" Else txtstudentid = "" txtstudentname.SetFocus txtstudentid.SetFocus txtstudentname = "" End If Else MsgBox txtstudentid + " is not a valid Student ID", "Graduation Project" txtstudentid = "" txtstudentname.SetFocus

> txtstudentid.SetFocus txtstudentname = "" End If

Else

txtstudentname = rs(0) + "" + rs(1)

End If

rs.Close

End Sub

Content Sub Command14_Click() Content GoTo Err Command14 Click

DoCmd.GoToRecord,, acFirst

Exit_Command14_Click: Exit Sub

Err_Command14_Click: MsgBox Err.Description Resume Exit_Command14_Click

End Sub

Private Sub Command15 Click()

Ce Error GoTo Err_Command15_Click

DoCmd.GoToRecord,, acPrevious

Exit_Command15_Click: Exit Sub

Err_Command15_Click: MsgBox Err.Description Resume Exit_Command15_Click

End Sub

Private Sub Command16_Click() On Error GoTo Err Command16_Click

DoCmd.GoToRecord,, acNext

Exit_Command16_Click: Exit Sub

Err_Command16_Click: MsgBox Err.Description Resume Exit Command16_Click

End Sub

Private Sub Command17_Click() On Error GoTo Err_Command17_Click

DoCmd.GoToRecord,, acLast

Exit_Command17_Click: Exit Sub

NEAR EAST UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

COM 400

GRADUATION PROJECT

LIBRARY AND BOOKSTORE AND DATABASE

SUBMITTED TO

MRS BESIME ERIN

SUBMITTED BY

IMRAN SOHAIL & BEDRIYE OZLEM PEKMEZ 960694 950204

Tables of Contents

1-Switch Board

•Switch Board for Departments and Currency

- Switch Board for Reports
 - Report of Book Issue
 - Report of Book Sale

2-Forms

Department Form •Codes for Form Department

Currency Form

•Codes for Form Currency

Book Form

•Codes for Form Books

Students Form

•Codes for Form Student

Book Issue Form

•Codes for Form Issue Book

• Codes for Form Issue Book II

Book Sale Form

• Codes for Form Book Sale

• Codes for Form Book Sale II Check Fine Form

• Codes For Form Check Fine

3- Conclusion

Acknowledgement:

First of all we are thankful to our God who is most kind and merciful. We are thankful to our parents who enabled us to do our graduation and do what we need, They did all those things we want. And at the and we want to say thanks to our friends who helped us in completion of our project. In the friends the most upper names are MUHAMMAD ATIF Malik, MUHAMMAD NAEEM,ILKER YIGIT, MEHET BOZKAN,IHSAN ZENGIN AND NECLA YIĞIT. We are also thankful to all the teachers who brought us at this point.

Introduction

Welcome to *Microsoft Access 2000*, In our Project you'll find out how to build applications that you and others can use to effectively manage data. In addition, you'll learn how to use Visual Basic® for Applications—the programming language included in Microsoft Access—to go beyond what you can accomplish by using only the Microsoft Access interface. With Visual Basic, you can extend and customize your application to fit your needs and those of your users.

If you're new to Microsoft Access, look first at Getting Results with Microsoft Access 2000 for information on creating the objects that make up a Microsoft Access database. In addition, consult the Office Assistant with questions that arise as you learn Microsoft Access.

Our Project has Four Sections:

The first section shows you how to tie database objects together into a coherent system and introduces Visual Basic for Applications, the programming language used in Microsoft Access. Examples in this section are based on the Orders sample application included with the Microsoft Access sample applications.

The second section, describes the elements of Visual Basic in more detail, including an overview of the objects and collections you can work within Visual Basic and the events you can respond to. In addition, it shows you how to use the Visual Basic debugging tools to test and debug your Visual Basic code, and explains how to respond to errors that occur while your code is running. The third section, shows you how to accomplish complex data management tasks in your application, including manipulating sets of records in code, developing an application for multiple users, communicating with other applications such as Microsoft Excel or Microsoft Word, and using library databases and dynamic-link libraries. In addition, the third section explains how to optimize, secure, and deliver your application to users.

The fourth section, which includes, explores powerful features that address special development needs, such as using ActiveX[™] controls, creating your own wizards and add-ins, working with external or client/server data, implementing database replication, and developing applications for the Internet and the World Wide Web.

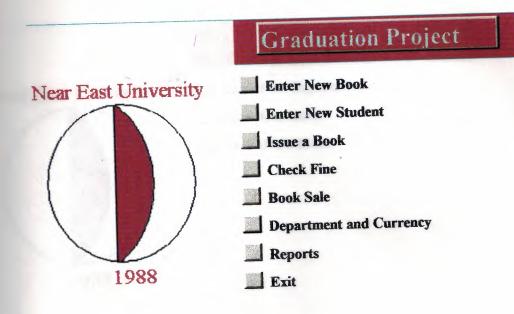
What's New in Microsoft Access 2000

Microsoft Access for Windows® 98, Windows NT and windows 2000 introduced many new features for application development. Microsoft Access 2000 introduces additional new features, as well as enhancements to existing features. The following sections highlight new features introduced in both versions.

Note If you have applications that were created with Microsoft Access 97 or earlier, you'll need to consider several important changes that affect your existing code and database objects. For more information, search the Help index for "converting databases."

Visual Basic for Applications Programming Language

Visual Basic for Applications replaces Access Basic, which was used in versions 2.0 and earlier, as the programming language for Microsoft Access. This is the same language you use to program in Microsoft Word, Microsoft Excel, and Microsoft Project, as well as the same language found in Microsoft Visual Basic.



Switch Board:

The first form, which we can see while opening the project, is the Switchboard. It contains all the buttons using them we can enter into the project and can work accordingly.

The main Items of switchboard are shown in the figure. It

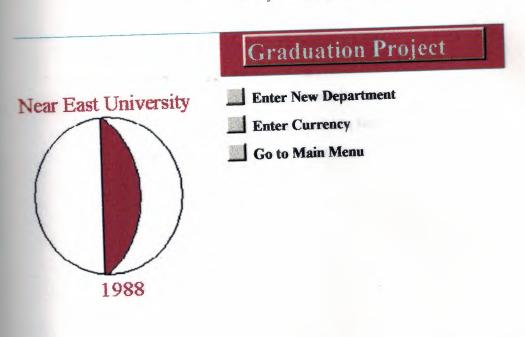
contains

- 1. Enter New Book
- 2. Enter New Student
- 3. Issue a Book
- 4. Check Fine
- 5. Book Sale
- 6. Department And Currency
- 7. Reports
- 8. Exit

Before going in detail of these all buttons, I want to tell you that in one switchboard we can add maximum 8 buttons, if we have more than 8 functions to do, So for this purpose we can add more than one switchboard, as we did here,

Caption "Department and Currency" is the switch to next switchboard, which consists of three fields

3



- 1. Enter New Department
- 2. Enter New Currency
- 3. Go to Main Menu

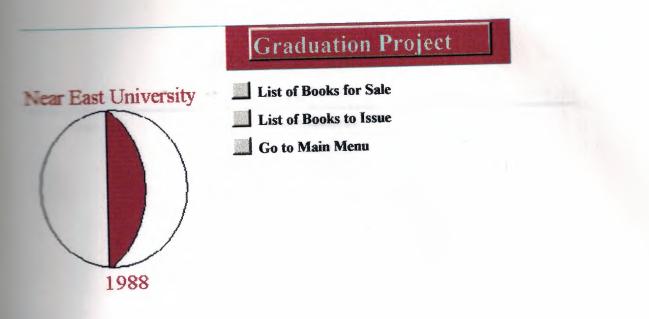
"Go to Main Menu" is the way back to the main menu or the first switchboard. The first button is to open the form, which is using to add new departments. And the same for the second, it will open the form, which is for the addition of new currency. We can see the detail of these forms later in the detail of forms.

If we have a look right down of this button there is "*Reports*" button which will guide us to the switch board of reports on which we have the buttons of reports clicking them we can see the reports.

Now if we fix a gaze on the other buttons of main switchboard, the first buttons we can see is the *"Enter New Book"*. This button leads us to the form for entering new books and the same for the others that every buttons is opening its related form or switchboard.

The last button is "*Exit*" to exit from the application or to end the program.

4



Switch Board for Reports.

Books For Sale

Author

Book Title

Hamlet

William Shakespeare

Quantity 20

Title Author	ISBN	Department
	Days (bern	
*		
in the second		
nin a sana sa mangana ang kangana da sana kang kang kang kang kang kang kang k		a line of the

Library and Book Store Database

ave
and a second sec

Form Department

		400 y 200 J 444 A 7744 4 1 104	eire'
1. 1. 1. 1. 1.			
and the second	<u>.</u>		
Ex	if		
	E	Exit	Exit

Form Currency

Forms:

The form is a screen view on which we can edit or add our new data entry. In our Project we have nine different forms, Names are listed in the switchboard. As we have described in the section of switchboard. Now we are going to give a detail of all forms individually.

We will describe it in the routine of usage. The first form we will use to add data of our need that is the department and currency forms. We will add some departments and currency using this form.

Department Form

In this form we have two fields and three command buttons. The first field is the department number, which is describing the serial number of departments and the second field is for the departmental names. The first command button is the "*New*" button, which is for adding a new department. When you will press this button a new number will appear in the department number field and you can write new name in the department name field. After doing this you have to press the second button, which is "*Save*" for saving the data you entered. And the third button, as its caption is showing that it is "*Exit*" button, is to exit from the form.

Codes for Form Department

Option Compare Database Option Explicit

Private Sub cmdsave_Click() Dim rssave As New ADODB.Recordset

rssave.Open "select * from tbldept where flddeptno = "' + Str(flddeptno) + "", CurrentProject.Connection, adOpenDynamic, adLockPessimistic, adCmdText

> rssave.AddNew rssave("flddeptno") = flddeptno rssave("flddeptname") = flddeptname

rssave.Update

rssave.Close

End Sub

Private Sub Command5_Click() On Error GoTo Err_Command5_Click

DoCmd.Close

Exit_Command5_Click: Exit Sub

Err_Command5_Click: MsgBox Err.Description Resume Exit_Command5_Click

Private Sub Cmdnew_Click() Dim rs As New ADODB.Recordset

rs.Open "select max(cint(flddeptno))+1,count(*) from tbldept", CurrentProject.Connection, adOpenDynamic, adLockPessimistic, adCmdText

If rs(1) = 0 Then flddeptno = 1 Else flddeptno = rs(0) End If flddeptname = ""

rs.Close End Sub

Codes for Form Currency

```
Option Compare Database
```

```
Private Sub Cmdnew_Click()
```

```
Dim rsnew As New ADODB.Recordset
```

```
rsnew.Open "select max(cint(fldcurid))+1,count(*) from tblcur",
CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic,
adCmdText
```

```
If rsnew(1) = 0 Then
fldcurid = 1
Else
'MsgBox rsnew(0)
fldcurid = rsnew(0)
rsnew.Close
End If
fldcurname = ""
```

End Sub

```
Private Sub cmdsave_Click()
Dim rssave As New ADODB.Recordset
rssave.Open "select * from tblcur ", CurrentProject.Connection,
adOpenForwardOnly, adLockPessimistic, adCmdText
rssave.AddNew
rssave("fldcurid") = fldcurid
rssave("fldcurname") = fldcurname
rssave.Update
rssave.Close
End Sub
```

Private Sub Command6_Click() On Error GoTo Err_Command6_Click

DoCmd.Close

Exit_Command6_Click: Exit Sub Err_Command6_Click: MsgBox Err.Description Resume Exit_Command6_Click

Currency Form

The currency form is working same as the department is, here also we have two fields and three command, the first field is for rency number to give the serial number of currency and the second is for rency name. And the rest of three command buttons are working as in *Department Form*.

Book Title	
Author ISBN	
1351	
Save	New Exit
	Book Form I
	Book Form I
Department	Book Form I
Department Language	Book Form I
	Book Form I
Language	

Book Form II

Book Form

As the name of the form is showing the characteristics of the form that this form is concerned with the books of library. So in this form have two parts name "Book Name" and "Book Detail". In Book Name have four main fields and three buttons. The first field, which has no label, is giving the number to count that how many books we have entered. The second field is "Book Title" which is for the Book Title. The third is for the Author Name and the last on is for the ISBN of that Book. The three command Buttons or working as the caption or showing the properties of them, The first one is New button whenever we want to add a book we have to press this button. The second one is for Saving the record, which we have entered but before pressing the Save button we have to complete the whole form including information of Book Detail.

In this Book Detail Form we have actually six fields but in picture we can see only three. The first field is a Combo Box, which have the list of all departments, which we have entered in the "Department Form". The purpose of this combo box is to give the identification to book that this book is of which department. The second field is the language field to show that the book is in which language.

The purpose of third check box, which is labeled, as "For Sale" is to show that whether this book is for sale or for issue, When you will check this button three more fields will appear Labeled as "Unit Price" "Currency" and "Quantity" respectively. In first text box we have to enter the price of that book, the second one, which is actually a combo box, is showing all the currencies, which we entered in "<u>Currency Form</u>". Here we will select that the price of book is in which currency. And the last one, which is "Quantity", it will show that what is the quantity of that specific book we have in our library for sale.

If we are checking this box the program will consider that this book is for issuing not for sale. After completing this information of the books we will go back to the "Book Name" side and we will press Save Button.

Codes for Form Books

Option Compare Database Option Explicit

Private Sub chkforsale_Click() If chkforsale = True Then txtqty.Visible = True txtprice.Visible = True txtprice.Enabled = True cmbcur.Visible = True Else txtqty.Visible = False txtprice.Visible = False txtprice.Enabled = False cmbcur.Visible = False End If End Sub

Private Sub Command10_Click() On Error GoTo Err Command10_Click

DoCmd.GoToRecord,, acFirst

Exit_Command10_Click: Exit Sub

Err Command10_Click:

MsgBox Err.Description Resume Exit_Command10_Click

End Sub

Private Sub Command11_Click() On Error GoTo Err Command11_Click

DoCmd.GoToRecord,, acPrevious

Exit_Command11_Click: Exit Sub

Err_Command11_Click: MsgBox Err.Description Resume Exit_Command11_Click

End Sub

Private Sub Command12_Click() On Error GoTo Err_Command12_Click

DoCmd.GoToRecord,, acNext

Exit_Command12_Click: Exit Sub

Err_Command12_Click: MsgBox Err.Description Resume Exit_Command12_Click

Command13_Click() Command13_Click

DoCmd.GoToRecord,, acLast

Exit_Command13_Click: Exit Sub Err_Command13_Click: MsgBox Err.Description Resume Exit_Command13_Click

End Sub

```
Private Sub Command14_Click()
Dim rsnew As New ADODB.Recordset
rsnew.Open "select max(cint(fldbookid))+1,count(*) from
blbookmain", CurrentProject.Connection, adOpenForwardOnly,
adLockPessimistic, adCmdText
```

If rsnew(1) = 0 Then fldbookid = 1 Else fldbookid = rsnew(0) rsnew.Close End If chkforsale = False fldbookname = "" fldauthor = "" fldauthor = "" fldlanguage = "" txtprice = "" txtprice = False txtprice.Enabled = False cmbcur.Visible = False

End Sub

Private Sub Command15_Click() On Error GoTo Err_Command15_Click

DoCmd.Close

Exit_Command15_Click: Exit Sub

Err_Command15_Click: MsgBox Err.Description Resume Exit_Command15_Click

End Sub

Private Sub Command18 Click() Dim rssave As New ADODB.Recordset rssave.Open "select * from tblbookmain ", CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic, adCmdText rssave.AddNew rssave("fldbookid") = fldbookid rssave("fldbookname") = fldbookname rssave("fldauthor") = fldauthor rssave("fldisbn") = fldisbn rssave("flddept") = flddept rssave("fldlanguage") = fldlanguage rssave("fldprice") = txtprice rssave("fldforsale") = chkforsale rssave("fldcur") = cmbcur rssave("fldqty") = txtqty rssave.Update rssave.Close

End Sub

Private Sub Form_Load() chkforsale = False txtqty.Visible = False

HOLE ROOM \$1

Students Form:

This form is to add the information of Students, that means that when a student is going to issue a book that has to register in the

	and the second second second
Student ID	950204
First Name	B.Pekmez
Last Name	Ozlem
Department	Computer Engineering
14 4 >	Ne D+

Student Form I

Address	Gazantepe
Tel. Phone	2258745
E- Mail	bedisp@hotmail.com

Student Form II

In this form we have seven fields labeled as "Student ID" First Name" "Last Name" "Department" "Address" "Telephone Number" E-Mail". The first field is the student ID, before entering any new student's monation we have to press new button, after this we will enter the ID of Student. As we will lose the focus from this field we will get a message from program, actually there are two types of messages. If the student ID is entering first time the program will give a message.

The program could not find the ID, Do you want to enter this ID"

There are two options Yes and No. If you will choose yes then the program will set the focus on the field "First Name". And you can continue with entering the information. And you will choose the second option that is NO the program will erase student ID to enter a new ID.

If you forget to enter a ID and you try to enter directly the name the program will give you and error message which is

You can not enter a name without entering ID"

This message will appear and on clicking ok the program will erase the name and will set the focus on ID. The rest of the fields are doing the normal operations as labeled on the form.

We have some extra buttons with some images on them they are for moving the data, the most left side button is for the "First Record" the next is for the Previous Record, The next to this button is for the "Next Record" and the fourth one is for "Last Record". The most right side button is the "Exit" button to close the form.

Codes for Form Student

Option Compare Database Option Explicit Dim a

Private Sub Command10_Click() On Error GoTo Err Command10 Click

a = 1 DoCmd.GoToRecord , , acNewRec

Exit_Command10_Click: Exit Sub

Err_Command10_Click: MsgBox Err.Description Resume Exit_Command10_Click

End Sub

Private Sub Command11_Click() On Error GoTo Err_Command11_Click

DoCmd.GoToRecord,, acFirst

Exit_Command11_Click: Exit Sub

Err Command11 Click:

MsgBox Err.Description Resume Exit_Command11_Click

End Sub

Private Sub Command12_Click() On Error GoTo Err_Command12_Click

DoCmd.GoToRecord,, acPrevious

Exit_Command12_Click: Exit Sub

Err_Command12_Click: MsgBox Err.Description Resume Exit_Command12_Click

End Sub

Private Sub Command13_Click() On Error GoTo Err_Command13_Click

DoCmd.GoToRecord , , acNext

Exit_Command13_Click: Exit Sub

Err_Command13_Click: MsgBox Err.Description Resume Exit_Command13_Click

Command14_Click() Command14_Click() Command14_Click

DoCmd.GoToRecord,, acLast

Exit_Command14_Click: Exit Sub

Err_Command14_Click: MsgBox Err.Description Resume Exit_Command14_Click End Sub Private Sub Command17_Click() On Error GoTo Err Command17 Click

DoCmd.Close

Exit_Command17_Click: Exit Sub

Err_Command17_Click: MsgBox Err.Description Resume Exit_Command17_Click

End Sub

Private Sub Form_Load() a = 0 End Sub

Private Sub txtstdfname LostFocus()

If IsNull(txtstudentid) Or txtstudentid = "" Then

MsgBox "You are not allowed to Enter a Student without his/her ID", vbCritical, "Graduation Project" txtstdfname = ""

txtstudentid.SetFocus End If End Sub

```
Dim rs As New ADODB.Recordset
Dim rsstd As New ADODB.Recordset
Dim choice
```

```
rs.Open "select fldstdfname,fldstdlname from tblstdmain where

dstudentid = "" + txtstudentid + """, CurrentProject.Connection,

dOpenForwardOnly, adLockPessimistic, adCmdText

rsstd.Open "select fldstudentid from tblstdmain",

CurrentProject.Connection, adOpenDynamic, adLockPessimistic,
```

adCmdText

If txtstudentid = rsstd(0) And a = 1 Then MsgBox "The student Number " + txtstudentid + " Already Exist ", vbExclamation, " Graduation Project" txtstdfname = "" txtstdfname = "" txtstudentid = "" txtstudentid = ""

```
Else
If rs.EOF = True Then
If Len(txtstudentid) = 6 Then
```

```
choice = MsgBox("The Program could not find: " +
txtstudentid + " Do You Want to Enter this ID", vbYesNo, "Graduation
Project")
```

If choice = 6 Then ' DoCmd.OpenForm "frmstdmain", acNormal

'txtstudentid = ""

'txtstudentid.SetFocus txtstdfname = "" txtstdlname = ""

txtstdfname.SetFocus Else txtstudentid = "" txtstdfname.SetFocus txtstdlname = "" txtstudentid.SetFocus txtstdfname = ""

End If

Else If txtstudentid = "" Then Exit Sub

Else

MsgBox txtstudentid + " is not a valid Student ID",

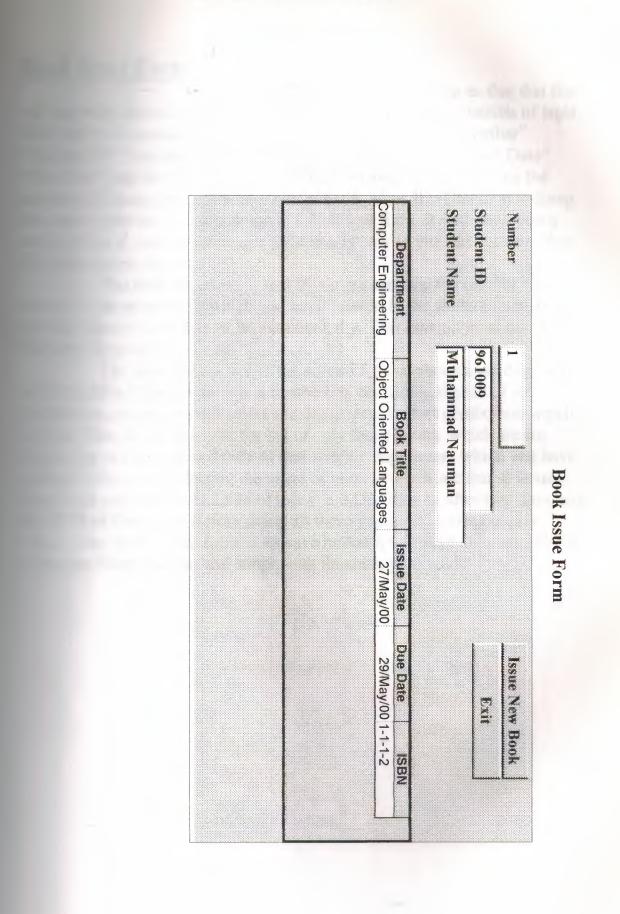
```
"Critical, "Graduation Project"
```

txtstudentid = "" txtstdfname.SetFocus txtstudentid.SetFocus txtstdfname = "" txtstdlname = "" End If End If Else

```
txtstdfname = rs(0)
txtstdlname = rs(1)
txtstudentid = ""
'txtstdfname.SetFocus
'txtstudentid.SetFocus
```

End If End If

rs.Close



Book Issue Form:

As the name of the form is showing us that this fine will use when some one is going to issue a book. This form consists of eight fields and two command buttons. The fields are labeled as "Number" Student ID" "Student Name" "Department" "Book Title" "Issue Date" Due Date" and the "ISBN". The first field Number is just showing the same operation as it was doing in the student ID. This field is doing the same operation as it was doing in the student form. It will show you a message box if you are entering a ID which is new or you did not enter that ID in the student form.

The difference here is that if you are entering the number which is already in the program or which you have entered in the student form, it will give the name of that student ID automatically. You have no need to enter the name by yourself.

The next thing which is the second form is showing as datasheet, its first field is Department, it is a combo box containing the list of all departments, when you will select the department in next combo box which is Book Title. It will give you the list of only those books, which are for Issuing but not for sale and only of that specific department which you have selected before. On Selecting the name of book you will see that in Issue Date Field you will see the date of today and Due date is After two days and the ISBN of that book. Before doing all these you have to press button "Issue New Book". And there is no save button it will save automatically as in Student Form. So you will simply exit finishing your work. Codes for Form Issue Book

Option Compare Database

Private Sub Cmdnew_Click() Dim rsnew As New ADODB.Recordset

DoCmd.GoToRecord , , acNewRec

rsnew.Open "select max(cint(fldissueno))+1,count(*) from bissuedet", CurrentProject.Connection, adOpenForwardOnly, bissuedet, adCmdText

> If rsnew(1) = 0 Then fldissueno = 1 Else

fldissueno = rsnew(0) rsnew.Close txtstudentid = "" txtstudentname = ""

End If

End Sub

Private Sub cmdnewstd_Click() DoCmd.OpenForm "frmstdmain", acNormal End Sub Private Sub cmdsave_Click()

Dem rssave As New ADODB.Recordset

rssave.Open "select * from tblissuedet", CurrentProject.Connection,

rssave("fldstudentid") = txtstudentid
rssave("fldstudentname") = txtstudentname
rssave.Update

rssave.Close

End Sub

Private Sub Form_Load() DoCmd.Maximize End Sub

```
Private Sub txtstudentid_AfterUpdate()
Dim rs As New ADODB.Recordset
Dim choice
    rs.Open "select fldstdfname,fldstdlname from tblstdmain where
distudentid = "" + txtstudentid + """, CurrentProject.Connection,
dOpenForwardOnly, adLockPessimistic, adCmdText
    If rs.EOF = True Then
        If Len(txtstudentid) = 6 Then
        choice = MsgBox("The Program could not find: " +
txtstudentid + " Do You Want to Enter this ID", vbYesNo, "Graduation
Project")
    If choice = 6 Then
```

DoCmd.OpenForm "frmstdmain", acNormal

txtstudentid = "" txtstudentname.SetFocus txtstudentid.SetFocus txtstudentname = "" Else txtstudentid = "" txtstudentname.SetFocus txtstudentid.SetFocus txtstudentname = "" End If Else MsgBox txtstudentid + " is not a valid Student ID", "Graduation Project" txtstudentid = "" txtstudentname.SetFocus

> txtstudentid.SetFocus txtstudentname = "" End If

Else

txtstudentname = rs(0) + "" + rs(1)

End If

rs.Close

End Sub

Content Sub Command14_Click() Content GoTo Err Command14 Click

DoCmd.GoToRecord,, acFirst

Exit_Command14_Click: Exit Sub

Err_Command14_Click: MsgBox Err.Description Resume Exit Command14 Click

End Sub

Private Sub Command15 Click()

Ce Error GoTo Err_Command15_Click

DoCmd.GoToRecord,, acPrevious

Exit_Command15_Click: Exit Sub

Err_Command15_Click: MsgBox Err.Description Resume Exit_Command15_Click

End Sub

Private Sub Command16_Click() On Error GoTo Err Command16_Click

DoCmd.GoToRecord,, acNext

Exit_Command16_Click: Exit Sub

Err_Command16_Click: MsgBox Err.Description Resume Exit Command16_Click

End Sub

Private Sub Command17_Click() On Error GoTo Err_Command17_Click

DoCmd.GoToRecord,, acLast

Exit_Command17_Click: Exit Sub Err_Command17_Click: MsgBox Err.Description Resume Exit_Command17_Click



End Sub

Private Sub Command19_Click() On Error GoTo Err_Command19_Click

DoCmd.Close

Exit_Command19_Click: Exit Sub

Err_Command19_Click: MsgBox Err.Description Resume Exit_Command19_Click

Codes for Issue Book II

Option Compare Database Option Explicit

Private Sub cmbbookid_AfterUpdate() Dim rs As New ADODB.Recordset Dim rslook As New ADODB.Recordset Dim rsret As New ADODB.Recordset

```
rs.Open "SELECT tblbookmain.fldisbn FROM tblbookmain where
fldbookname = "' + cmbbookid + "'", CurrentProject.Connection,
adOpenForwardOnly, adLockPessimistic, adCmdText
    fldisbn = rs(0)
    txtissuedate = Date
    txtduedate = txtissuedate + 2
'fldstudentid = ""
    rs.Close
    rs.Open "select * from tblissuebookdet where fldbookid = "' + cmbbookid
+ ""', CurrentProject.Connection, adOpenForwardOnly, adLockOptimistic,
adCmdText
```

If **rs**.EOF = False Then

If rs(6) = True Then

rs.MoveLast rs("flddepartment") = cmbdepartment rs("fldbookid") = cmbbookid rs("fldisbn") = fldisbn rs("fldissuedate") = txtissuedate rs("fldduedate") = txtissuedate + 2 rs("fldret") = False rs.Update

Else

rslook.Open "select max(fldisbn),max(fldbookid),max(fldduedate),count(*) from tblissuebookdet where fldbookid = "" + cmbbookid + """, CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic, adCmdText

If IsNull(fldisbn) Then

Exit Sub

```
rslook.Close
 Else
If rslook(1) = cmbbookid Then
```

If rslook(2) > Date Then

MsgBox "The Book is Issued to Some other

Student, Due date of the book is" & rslook(2), vbExclamation, "Graduation Project"

'DoCmd.DoMenuItem acFormBar, acEditMenu, 8,,

acMenuVer70

'DoCmd.DoMenuItem acFormBar, acEditMenu, 6, ,

acMenuVer70

'On Error Resume Next

```
cmbbookid.SetFocus
       txtissuedate = ""
       txtduedate = ""
       fldisbn = ""
        Else
      Exit Sub
      End If
End If
End If
```

rs.Close

End Sub

End If End If

Private Sub cmbdepartment_AfterUpdate() cmbbookid.RowSource = "SELECT tblbookmain.fldbookname FROM tblbookmain where flddept = "' + cmbdepartment + "' and fldforsale = false"

		on the sea.	
Number			
Student ID			
Book Name			
Date of Return			
Issue Date			
Due Date			
Nev	x Save	Exit	

Form Check Fine

Form Check Fine:

In this form we have six fields labeled as "Number" "Student ID" "Book Name" "Date of Return" "Issue Date" "Due date" The first field is giving only the number. The student ID field is to enter the ID of student, which is coming to return the date, when you will enter the ID it will give the Book Name automatically, which that student has issued. And also the issue date and the due date. You will enter the return date and when you will go out of this field if the return date is same as the due date it will give a message box.

"No Charge"

and if the return date is greater than the due date it will calculate the delayed days and calculate the fine the default value of fine is 50,000TL so it will calculate according to this value.

The command button Save, if you will press this button it means that this book has returned and the money is paid. Other wise the book is not returned and the also the fine is not paid.

Codes for Form Check fine

Option Compare Database Option Explicit

Private Sub Cmdnew_Click()

Dim rsnew As New ADODB.Recordset rsnew.Open "select max(cint(fldissue))+1,count(*) from tblchkfine", CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic, adCmdText

If rsnew(1) = 0 Then txtissue = 1 Else txtissue = rsnew(0) rsnew.Close End If clearall

End Sub

Private Sub cmdsave_Click() saveall End Sub

Private Sub Command14_Click() On Error GoTo Err_Command14_Click

DoCmd.Close

Exit Command14_Click:

Exit Sub

Err_Command14_Click: MsgBox Err.Description Resume Exit_Command14_Click

End Sub

Private Sub clearall() fldstudentid = "" fldbookname = "" fldissuedate = "" fldduedate = "" fldreturndate = "" End Sub

Private Sub saveall()

Dim rssave As New ADODB.Recordset rssave.Open "select * from tblchkfine ", CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic, adCmdText rssave.AddNew rssave("fldissue") = txtissue rssave("fldstudentid") = fldstudentid rssave("fldbookname") = fldbookname rssave("flduedate") = fldissuedate rssave("flduedate") = fldduedate rssave("fldreturndate") = fldreturndate rssave.Update rssave.Close rssave.Close rssave.Open "select fldret from tblissuebookdet where fldbookid= "' + fldbookname + "' ", CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic, adCmdText

rssave(0) = vbYes rssave.Update

End Sub

```
Private Sub fldreturndate_LostFocus()
Dim rsfine As New ADODB.Recordset
Dim ret
Dim days As Variant
Dim fine As Variant
rsfine.Open "select
```

max(fldduedate),max(fldissuedate),max(fldret),max(days),count(*) from qryissue where fldstudentid = "' + fldstudentid + """, CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic, adCmdText

```
If (rsfine(1) >= fldreturndate) Then

MsgBox "No Charge"

Else

days = fldreturndate - rsfine(0)

If days = 0 Then

MsgBox "No Charge", vbInformation, "Graduation Project"

Else

fine = days * 50000

MsgBox "You are " + "" & days & "" + " days Late and

you Have to Pay:" & " " & fine & " TL as Fine", vbCritical, "Graduation

Project"
```

End If

End If rsfine.Close End Sub

Private Sub fldstudentid_LostFocus() Dim rs As New ADODB.Recordset 'Dim rsdet As New ADODB.Recordset ' rsdet.Open "select * from tblissuebookdet where fldbookname = "' + fldbookname + "", CurrentProject.Connection, adOpenDynamic, adLockPessimistic, adCmdText

rs.Open "select * from qrychkfine where fldstudentid = "" + fldstudentid + " and fldret = false", CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic, adCmdText

If rs.EOF = True Then

MsgBox fldstudentid + " " + "Have not Issued any Book", vbInformation, "Graduation Project" clearall

'fldbookname.SetFocus 'fldstudentid.SetFocus

Exit Sub

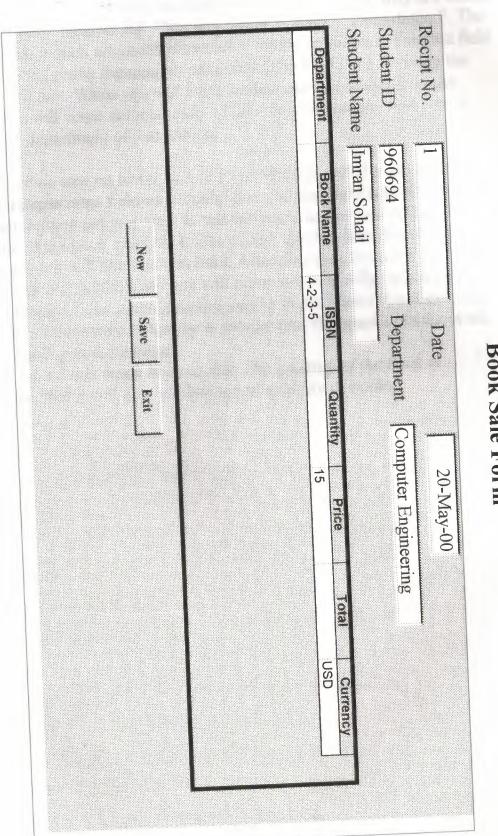
Else

ŧ

Do While rs.BOF = False fldbookname = rs(1)fldissuedate = rs(2)fldduedate = rs(3)fldreturndate = "" Loop 1

End If

rs.Close End Sub



Book Sale Form

Book Sale Form:

In this form we have twelve fields in total, they are labeled as "Receipt No" "Student ID" "Student Name" " Date" "Department". The first five fields, which are listed above are working as follows. The first field "Receipt No" is giving the number of receipt. The field Date is giving the date automatically. When you will enter student number in this field the student name will come automatically of that student number. And also it will give the department of that student.

If we saw on lower part of form which is called a sub form we have also a department field as a combo box, but this field is to choose that form which department you want to sell the book, and as you will choose the department, in the book field the books of that specific department and those which are for sell will come in this field, After choosing the book the ISBN, the Price of Book and the Currency will come automatically. When you enter the quantity if that much quantity will be in the library it will calculate the total, and if the entered quantity is greater than the quantity in the stock of library it will give a message

"You Don't have this much book to sell, The quantity of the book is" That's means that it will give you the actual quantity of books.

Codes for Form Book Sale

Option Compare Database Option Explicit

Private Sub cmdclose_Click() On Error Resume Next DoCmd.Close End Sub

Private Sub cmdclose_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single) cmdclose.ForeColor = vbRed End Sub

Private Sub Cmdnew_Click() Dim rs As New ADODB.Recordset

DoCmd.GoToRecord,, acNewRec

```
rs.Open "select max(cint(fldreciptno))+1,count(*) from tblsalemain ",
CurrentProject.Connection, adOpenDynamic, adLockPessimistic,
adCmdText
If rs(1) = 0 Then
```

```
txtreciptno = 1
Else
txtreciptno = rs(0)
End If
txtdate = Date
```

'Forms!formsalemain!formsaledet.Form!cmbbookid = ""

rs.Close

Private Sub cmdsave_Click() Dim rssave As New ADODB.Recordset

rssave.Open "select * from tblsalemain where fldreciptno = " + Str(txtreciptno) + "", CurrentProject.Connection, adOpenDynamic, adLockPessimistic, adCmdText

If rssave.EOF = False Then rssave("fldreciptno") = txtreciptno rssave("flddept") = cmbdept rssave("flddate") = txtdate

rssave.Update End If rssave.Close

End Sub

Private Sub Detail_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single) cmdclose ForeColor = vbBlue End Sub

Private Sub txtstudentid LostFocus() Dim rs As New ADODB.Recordset

Dim choice

rs.Open "select fldstdfname,fldstdlname,flddeptname from qrystdbydept where fldstudentid = "" + txtstudentid + """, CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic, adCmdText

If rs.EOF = True Then

If Len(txtstudentid) = 6 Then choice = MsgBox("The Program could not find: " + txtstudentid + " Do You Want to Enter this ID", vbYesNo, "Graduation

Project")

If choice = 6 Then DoCmd.OpenForm "frmstdmain", acNormal

txtstudentid = "" txtstudentname.SetFocus txtstudentid.SetFocus txtstudentname = "" txtdept = "" Else txtstudentid = "" txtstudentname.SetFocus txtstudentid.SetFocus txtstudentname = "" txtdept = "" End If Else MsgBox txtstudentid + " is not a valid Student ID", vbCritical, "Graduation Project" txtstudentid = "" txtstudentname.SetFocus txtstudentid.SetFocus txtstudentname = "" txtdept = "" End If Else txtstudentname = rs(0) + "" + rs(1)txtdept = rs(2)End If rs.Close

End Sub

Private Sub Form_Load()

DoCmd.Maximize Me.Refresh

End Sub

Codes for Form Book Sale II

Option Compare Database Option Explicit

Private Sub cmbbookid AfterUpdate() Dim rs As New ADODB.Recordset

rs.Open "select fldbookname, fldisbn, fldprice, fldcurname from qrybooksale where fldbookid = " + cmbbookid + " and fldforsale = true", CurrentProject.Connection, adOpenForwardOnly, adLockPessimistic, adCmdText

```
txtisbn = rs(1)
txtprice = rs(2)
txtcur = rs(3)
txtqty = ""
txttotal = ""
```

rs.Close

End Sub

Private Sub cmbdept_AfterUpdate()

cmbbookid.RowSource = "SELECT distinct fldbookid, fldbookname FROM qrybooksale where flddept = " + cmbdept + " and fldforsale = true " cmbbookid = Forms!formsalemain!formsaledet.Form!cmbbookid.ItemData(0) If cmbbookid.ListCount = 0 Then

MsgBox "There is No Book for sale", vbCritical, "Graduation Prject"

Exit Sub End If

End Sub

Private Sub Form_Load() Call cmbdept_AfterUpdate End Sub

```
Private Sub txtqty_LostFocus()
Dim rscount As New ADODB.Recordset
Dim rs As New ADODB.Recordset
Dim b
    rs.Open "select * from tblbookmain where fldbookid = " + cmbbookid
```

+ "", CurrentProject.Connection, adOpenDynamic, adLockPessimistic, adCmdText

rscount.Open "select max(fldqty),count(*) from tblbookmain where fldbookid = "' + cmbbookid + "' and fldforsale = true",

CurrentProject.Connection, adOpenDynamic, adLockPessimistic, adCmdText

'MsgBox rscount(0)

```
b = rscount(0) - txtqty
```

If b < 0 Then

MsgBox "You Dont have this much Books to sell. Books for sale are " + rscount(0) + " ", vbCritical, "Graduation Project"

```
txtqty = ""
txttotal = ""
txttotal.SetFocus
txtqty.SetFocus
```

Else

```
txttotal = (Val(txtqty) * Val(txtprice))
rs("fldqty") = b
rs.Update
```

End If

rscount.Close rs.Close

```
End Sub
```

Conclusion:

In Microsoft Access 95, Visual Basic for Applications replaced Access Basic, which was used in versions 1x and 2.0 of Microsoft Access. In most respects, Visual Basic is identical to Access Basic, and Microsoft Access makes most necessary conversions to your code automatically when you convert your database.

However, the conversion process makes some changes to your code that you need to be aware of, and there are some additional changes that you must make yourself in order for your application to run successfully in Microsoft Access 2000.

Note The /runtime command-line switch is not available in the stand-alone or Office Professional Edition versions of Microsoft Access 2000. However, the /runtime command-line switch is available in the Office Developer Edition version of Microsoft Access 2000.

Calling the Windows Application Programming Interface

If your existing version 1.x or 2.0 Access Basic code makes calls to the <u>Windows application programming interface (API)</u>, you may need to modify these calls when you convert your database. Versions 1.x and 2.0 of Microsoft Access are 16-bit applications and run on 16-bit versions of Windows. Microsoft Access 2000 is a 32-bit application and runs on 32-bit versions of Windows \Box Windows 95 and the Windows NT platform.

Check any <u>Declare</u> statements to ensure that they refer to the correct 32-bit dynamic-link libraries (DLLs). For example, 32-bit versions of Windows include User32.dll; in earlier versions of Windows, this DLL is called User.dll.

The names of some functions in the Windows API have changed. Additionally, functions in the 32-bit Windows API are case-sensitive.

Some Properties and Optional Arguments Are Strictly Typed

Certain properties that returned Variant values in previous versions of Microsoft Access now return strings. Code that includes these properties will now run faster; however, when you convert a database from a previous version of Microsoft Access to Microsoft Access 2000, you should verify that no errors are introduced by these new return values.

In addition, some optional arguments which were of type Variant in previous versions of Microsoft Access are now strictly typed. Affected methods, properties, and functions include the GoToPage method of a Form object, the OpenCurrentDatabase, Echo, Quit, CreateForm, CreateReport, CreateControl, and CreateReportControl methods of the Application object, the Column property of a combo box or list box, and the domain aggregate functions. If your code includes any of these elements, you may want to verify that no errors are introduced when you convert your database.

To check for errors, first compile all code in the database. Next check for run-time errors when you run your code. The errors most likely to occur are a "Type mismatch" error, or an "Invalid use of Null" error.

You should also check for silent errors that may be introduced into your code when you convert a database. For example, the **RecordSource** property of a control previously returned a **Variant** value, but now returns a string. A string can't contain a **Null** value, although a **Variant** can. Code in a converted database that checks whether the value of the **RecordSource** property is **Null** will no longer run properly, because the value of the **RecordSource** property will never be **Null**. The following procedure is an example of code that checks whether the value of the **RecordSource** property of a text box is **Null**.

Private Sub Form_Load()
 If IsNull(Me.RecordSource) Then
 Me.Recordsource = "Employees"
 End If
End Sub

In Microsoft Access 2000, the IsNull function will never return True for the value of the RecordSource property. If the RecordSource property has not already been set in the previous example, its value is an empty string, the value returned by the IsNull function is False, and the procedure will never enter the If...Then construct to set the RecordSource property.

To correct the code in the previous example, use the Len function to check the length of the string returned by the **RecordSource** property to determine whether it is an empty string. The following line corrects the code in the previous example:

If Len(Me.RecordSource) = 0 Then

Changes to Modules

In versions 1.x and 2.0 of Microsoft Access, modules containing general procedures not specific to any form or report module are called global modules. In Microsoft Access 2000, these are called <u>standard</u> modules.

Microsoft Access 2000 includes <u>class modules</u>, which you can use to create custom objects. Any public procedures in a class module become methods and properties of a custom object when you create a new <u>instance</u> of the class. Class modules exist both independently and in association with forms and reports.

The CurrentDb Function Versus DBEngine(0)(0)

Use the CurrentDb method instead of

DBEngine(0)(0) to return a <u>Connection</u> object variable that points to the current database. The <u>CurrentDb</u> function creates another instance of the current database, whereas DBEngine(0)(0) refers to the open copy of the current database. Using DBEngine(0)(0) limits your ability to use more than one variable of type **Database** that refers to the current database.

The DBEngine(0)(0) syntax is still supported, so Microsoft Access doesn't change your code during the conversion process. However, you should consider making this modification to your code in order to avoid possible conflicts in a multiuser environment.

The CurDir Function

The <u>CurDir</u> function behaves differently than it does in version 1.x or 2.0 due to the way that applications interact with Windows 95. Since each application has its own current folder, setting the current folder in Windows 98 by double-clicking an icon doesn't affect the current folder in Microsoft Access. The **CurDir** function in Microsoft Access 2000 always returns the current path.

Assigning an OLE Object to a Variable

If you manipulate <u>OLE objects</u> or other binary data in your code, you should use an array of bytes to store binary data. In versions 1.x and 2.0 of Microsoft Access, you assign OLE objects, or other binary data less than 64K in size, to string variables when you need to manipulate the objects or data in code. You also assign the data returned by the <u>GetChunk</u> method to string variables. However, Visual Basic supplies a <u>Byte</u> data type and byte functions such as <u>LeftB</u> and <u>RightB</u>. In Microsoft Access 2000, you should store binary data in an array of bytes instead of a string variable, and you should use the byte functions to manipulate that data.

Declare DDE Channels as Variant or Long

If you use the **DDEInitiate** function to open a DDE channel, you can declare the variable that stores the channel number, which is a **Long** value, as either a **Variant** or **Long** value. In versions 1.x and 2.0 of Microsoft Access, the channel number is an **Integer** value, so you must modify any Declaration statements in your code that create variables of type **Integer** to store the channel number.

String Conversion Fails If the String Contains a Percent Sign (%)

You can't assign a string containing a percent sign to a variable or a field that has a numeric data type, as in the following example:

```
Dim intX As Double
intX = "10"
intX = "10%"
error.
```

' This works. ' This returns an

The hWnd Property

If you use the <u>hWnd</u> property in your code to pass a window handle of a form or a report to a Windows routine, you should pass the value directly to the routine. You shouldn't assign the value of this property to a variable. For example:

```
If Not IsZoomed(Screen.ActiveForm.hWnd) Then
DoCmd.Maximize
EndIf
```

In versions 1.x and 2.0 of Microsoft Access, the **hWnd** property of a form or report is an **Integer** value. In Microsoft Access 2000, the **hWnd** property is a **Long** value, which you must change your code to accept.

DAO Property Objects vs. Microsoft Access Property Objects

You can't use an object variable in your code to refer to a **Category** property. The **Category** property is no longer supported for <u>Form</u>, <u>Report</u>, and <u>Control</u> objects.

The Parent Property

In Microsoft Access 2000, if you use the <u>Parent</u> property of a <u>control</u> in code or in an <u>expression</u> on a form or report, it typically returns the Form or **Report** object that contains the control. For example, if CategoryID is a <u>text</u> box on the Categories form, the code

Forms!Categories!CategoryID.Parent returns a reference to the Categories form. There are two exceptions: for attached <u>labels</u>, the **Parent** property now returns the control the label is attached to; and for controls in an <u>option group</u>, it now returns the option group control.

Setting a Reference to a Microsoft Access Database

You cannot set a reference to a database created with a prior version of Microsoft Access from Microsoft Access 2000. You must convert the database to a Microsoft Access 2000 (version 9.0) database in order to set a.

Exclamation Point vs. Dot Operator When Referencing DAO Fields

If you used the . (dot) operator syntax when referring to a <u>Field</u> object of a <u>Recordset</u> object in applications created in versions 1.x or 2.0 of Microsoft Access, you must modify those references to use the ! (exclamation point) operator syntax. Or, if you want to continue using the . (dot) operator syntax, you must establish a reference to the Microsoft DAO 2.5/3.5 compatibility library in the **References** dialog box, available by clicking **References** on the **Tools** menu while in module <u>Design view</u>.

Converted ActiveX Controls

If your application contains <u>ActiveX controls</u> (formerly called OLE, or custom, controls) that were set up in version 2.0, you may need to insert the <u>ByVal</u> keyword in front of arguments that are passed to event procedures called from ActiveX control events, as in the following example:

Sub ChangeMonth_Click(ByVal intCurrentYear As Integer)

In order to determine whether an argument needs to be passed by value, click **Compile All Modules** on the **Debug** menu in module Design view. If you receive the following error message, you need to insert the **ByVal** keyword in front of the argument: "Event procedure declaration doesn't match description of event having the same name."

Because type checking of arguments is improved in Microsoft Access 2000, new event procedures created for ActiveX controls automatically have the **ByVal** keyword inserted when it's needed.

Calling Procedures in Form and Report Modules

In versions 1.x and 2.0 of Microsoft Access, you couldn't call a procedure defined in a form or report from anywhere but within that form or report module. In Microsoft Access 2000, you can call a public procedure in a form or report module from any procedure in the current database. You must qualify the procedure with the <u>class name</u> of the form or report module. For instance, to call a procedure named DisplayMessage that's defined in the module of the Orders form, you would use the following syntax:

Form Orders.DisplayMessage

It's better to place procedures that you will call from outside a form or report in a <u>standard module</u> if possible, rather than in a form or report module.

Referring to Forms, Reports, and Controls in the Debug Window

When you're testing and debugging code, you must fully qualify all references to objects that you use in the <u>Debug window</u>, unless you've suspended execution in a form or report module. This means that in the <u>Immediate pane</u> of the Debug window, you must use the code Forms!Categories!CategoryID to refer to the CategoryID control on the Categories form in Form view, instead of just CategoryID, even when the Categories form is the current form.

Also, you can't use the <u>Me</u> keyword in the Debug window to refer to an object on a form or report when that form or report is in Design view unless you've suspended the execution of code in the form or report.

Converting Forms and Reports That Contain ActiveX Controls

When you convert a Microsoft Access 2.0 database for use in Microsoft Access 2000 (version 9.0), ActiveX controls on forms and reports may not be converted automatically. Microsoft Access 2.0 supports 16-bit OLE controls, while version 9.0 supports only 32-bit ActiveX controls.

If you're converting a database that contains a form or report that has a 16bit version of an ActiveX control, and the 32-bit version doesn't yet exist on your system, Microsoft Access will generate an error message. You must obtain a 32-bit version of each ActiveX control that you wish to update and register it in the system registry. You should then save the form or report in the converted database and close and reopen the database.

Enabling a Database Created with a Previous Version of Microsoft Access

When you <u>enable</u> a large database created with version 1.x or 2.0 of Microsoft Access, but don't convert it to Microsoft Access 2000, you may need to increase the maximum buffer size beyond the default. To change this setting, open the system registry and navigate to the \HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Jet\3.0\Engines\Jet 2.x subkey. Create a new subkey named ISAM, and within this subkey create a new DWORD value called MaxBufferSize. Set this value to 1024, decimal base. For more information about editing the registry, see your Microsoft Windows documentation.

Time Values in Query Criteria

When you convert a version 1.x or 2.0 database to Microsoft Access 2000, queries that contain criteria based on specific time values in Date/Time fields may return different results than they do in earlier versions. This behavior may also occur if you link tables from a version 1.x or 2.0 database to a Microsoft Access 2000 database. Only the time portion of Date/Time fields is affected.

Visual Basic Functions That Are Not Supported in Expressions

The following six Visual Basic functions can't be used in expressions outside a user-defined **Sub** or **Function** procedure:

EOFLocFileAttrLOFFreeFileSeek

If you need to use one of these functions in an expression outside a procedure, you can call the function from within a user-defined function that you call from the expression.