



Non-Procedural

Object

Module

Application

Development

System

Lab Manual

Version 4.11
September 2001

Lab 1

Objective

The objective of this lab is to create a GUI application to maintain a file. This involves setting up a new library, creating a panel with assorted controls, attaching code to the controls, and grouping controls to be enabled and disabled.

Description

In this lab, you will create a new library called **wrklib.en** to be used in the Labs. Then you will create a panel called SALESMAINT and a program called SALEPROG to maintain a file called SAL_MAST which contains data pertaining to the people in the Sales Department.

The maintenance panel will look like this:

The screenshot shows a window titled "Salesperson Information". It contains the following fields and controls:

- Salesperson ID: ABC
- Name: Alfred B. Carter
- Commission:
- Commission Rate (%): 10
- Sales to date: \$123,456,789.00

At the bottom of the window, there are four buttons: a green plus icon, "Write", a green plus icon, and "Delete", followed by a red X icon and "Exit".

The SAL_MAST file consists of a key field and four data fields as follows:

SAL_ID (KEY)	A unique three character code (uppercase) to identify the salesperson
SAL_NAME	The name of the salesperson
SAL_COMM	A flag containing "Y" or "N" to indicate whether the salesperson receives a commission
SAL_COMM_RATE	The commission rate
SAL_TOTAL	Total annual sales to date.

Lab 1.1: Creating a New Library

Create a new library to be used by all the Lab exercises.

1. Start a new PVX session.
2. RUN “*NOMADS”
3. Create a new library. Select *Library/New* from the NOMADS II menu. Enter **wrklib.en** as the library name and press ‘OK’.
4. Set library defaults. Press the Library Defaults button on the Library Object Selection panel. Add an appropriate description for the library. Then select the Font/Clr folder and set font to MS Sans Serif, Foreground colour to Black, Background colour to Light Gray, and check the Underscore &’ed characters attribute. Press ‘OK’.

Lab 1.2: Creating a Maintenance Panel

Create a panel to be used to maintain the SAL_MAST file.

1. Create a panel called SALESMAINT. Enter **SALESMAINT** in the Name input field of the Library Object Selection screen and press the button marked ‘Panel Object’.
2. Define the panel characteristics. Select the Header option by pressing the button marked ‘Head’ on the toolbar.
 - (a) In the Display folder, enter **Salesperson Information** in the title input box, and enter **SALEPROG** as the default program. Set width to 70 and height to 16.
 - (b) In the Attributes folder, check Dialogue and Suppress.VAL.
 - (c) Press ‘OK’.

Lab 1.3: Drawing Controls

Add the controls which will be needed to create the user interface for the SALESMAINT application. These will include multiline input boxes for the salesman id, name, commission rate and total sales to date, and a check box to flag if the salesman is to receive a commission. There will also be buttons to write and delete records and an exit button.

1. Add the text prompts for the multilines. Select the fonted text control from the tool box, then use the mouse to position the text on the panel.

(a) In the Display folder, enter **Salesperson &ID:** in the text input field, and adjust the position to column 3, line 2, and the size to a width of 25 and a height of 1. Press 'OK'

(b) Repeat step (a) three more times using the following criteria:

Name	Title	Column	Line	Width	Height
FONTED_TEXT_2	&Name:	3	4	25	1
FONTED_TEXT_3	Commission &Rate:	3	8	25	1
FONTED_TEXT_4	&Sales to Date:	3	10	25	1

2. Add a multiline for SAL_ID. Select the multiline control from the tool box and position it on the panel next to Salesperson ID. Enter **SAL_ID** into the Name input field.

(a) In the Display folder, enter the following values:

Input length: 3

Column: 30

Line: 2

Wide: 6

High: 1

(b) In the Attributes folder, check Upper case and set Alt-key to I (as in ID).

(c) Press 'OK'.

3. Add a multiline named **SAL_NAME**.

(a) In the Display folder, enter the following values:

Length: 40

Column: 30

Line: 4

Wide: 35

High: 1

(b) Press 'OK'.

4. Add a check box named **SAL_COMM**.

(a) In the Display folder, enter **&Commission** in the Text box and check Button on Right. Set column to 3, line to 6, width to 29 and height to 1.

(b) In the Values folder, set the Default Initial value to **N**, and set the translation values by setting the Off State to **N** and the On State to **Y**. Press 'OK'.

5. Add a multiline named **SAL_COMM_RATE**.

(a) In the Display folder, enter the following values:

Input length: 2

Input format: #0

Column: 30

Line: 8

Wide: 4

High: 1

(b) In the Attributes folder, check Numeric, Initially disabled, and set the Alt-key to R.

(c) In the Validation folder, enter **0-99** in Rules.

(d) Press 'OK'

6. Add a multiline named **SAL_TOTAL**.
 - (a) In the Display folder, enter the following values:
 Input length: 15
 Input format: \$###,###,##0.00
 Column: 30
 Line: 10
 Wide: 15
 High: 1
 - (b) In the Attributes folder, check Numeric and set the Alt-key to S.
 - (c) In the Validation folder, enter **0-999999999.99** in Rules.
 - (d) Press 'OK'
7. Add the Write button. Select the Button control from the tool box and place it on the panel. Name the button **WRITE**. Enter **&Write** in the text field. Then select a bitmap by pressing the 'Browse' button next to 'Normal', clicking an appropriate bitmap (such as !File_add) and pressing 'OK'. Set the column to 3, line to 13, width to 15 and height to 2.
8. Add a button named **DELETE**. Enter **&Delete** in the text field. Select an appropriate bitmap (such as !File_del). Set the column to 18, line to 13, width to 15 and height to 2.
9. Add a button named **EXIT**. Enter **E&xit** in the text field and select a bitmap (such as !X). Set the column to 52, line to 13, width to 15 and height to 2. In the Logic folder, open up the 'When Button Pressed' drop box and select END.
10. Press the button with the diskette icon on the toolbar to write the control descriptions to file.
11. Press the 'TEST' button on the tool bar to test data entry for the various controls. Exit test mode by pressing the Exit button.

Lab 1.4: Attaching Code to Controls

Write and attach the code to drive the SALESMAINT application.

1. Add the code references to the Logic folders of the panel and controls:
 - (a) Open the panel Header definition and select the Logic folder. For 'Pre-display' logic, select 'Perform' and enter **“;Initialize”** in the input field. For 'On exit' logic, select 'Perform' and enter **“;Wrapup”** in the input field.
 - (b) Open the Logic folders of the following controls and set up the logic as follows:

SAL_ID	On Select	Perform “;Process_SAL_ID”
SAL_COMM	When Button Pressed	Perform “;Process_SAL_COMM”
WRITE	When Button Pressed	Perform “;Process_WRITE”
DELETE	When Button Pressed	Perform “;Process_DELETE”
2. Record the changes by pressing the save button on the toolbar.
3. Start a new PVX session to create the data file and to write the code which is to be attached to the controls.
4. Create the SAL_MAST keyed file by entering the following command:
KEYED “SAL_MAST”,3

5. Enter the following program:

```
00010 ! saleprog - Salesperson Maintenance Program
00020 !
00030 INITIALIZE:
00035 SAL_NAME$="", SAL_ID$="", SAL_COMM$="N", SAL_COMM_RATE=0, SAL_TOTAL=0
00040 LET SM_FN=HFN; OPEN (SM_FN) "SAL_MAST"
00050 RETURN
00060 !
00070 PROCESS_SAL_ID:
00080 IF SAL_ID$="" THEN GOTO *RETURN
00090 REFRESH_FLG=1
00100 READ (SM_FN,KEY=SAL_ID$,DOM=NEW_REC) IOL=1000
00110 RETURN
00120 NEW_REC:
00130 LET SAL_NAME$="", SAL_COMM$="N", SAL_COMM_RATE=0, SAL_TOTAL=0
00140 RETURN
00150 !
00160 PROCESS_SAL_COMM:
00170 RETURN
00180 !
00190 PROCESS_WRITE:
00200 IF SAL_ID$="" THEN MSGBOX "Missing Salesperson Code","Error"; LET
:NEXT_ID=SAL_ID.CTL; GOTO *RETURN
00210 WRITE (SM_FN,KEY=SAL_ID$) IOL=1000
00220 MSGBOX "Record "+SAL_ID$+" updated","Update"
00230 LET NEXT_ID=SAL_ID.CTL
00240 RETURN
00250 !
00260 PROCESS_DELETE:
00270 REMOVE (SM_FN,KEY=SAL_ID$,DOM=NO_SUCH_REC)
00280 MSGBOX "Record "+SAL_ID$+" deleted","Delete"
00290 LET NEXT_ID=SAL_ID.CTL
00300 RETURN
00310 NO_SUCH_REC:
00320 MSGBOX "The record "+SAL_ID$+" does not exist","Delete"
00330 RETURN
00340 !
00350 WRAPUP:
00360 CLOSE (SM_FN)
00370 RETURN
00380 !
01000 IOLIST SAL_NAME$, SAL_COMM$, SAL_COMM_RATE, SAL_TOTAL
```

6. Save the program as "saleprog".

Lab 1.5: Assigning Groups to Controls

Assign a group to the SAL_COMM_RATE control so that it can be enabled/disabled depending on whether SAL_COMM is checked or not.

1. Press Alt-Tab to return to the NOMADS designer session.
2. Make the SAL_COMM_RATE multiline part of the COMMISSION group. Open the SAL_COMM_RATE definition and press the 'Groups' button. Enter COMMISSION in the 'New' input field and press the 'New' button. Press 'OK'.
3. Record the changes by pressing the save button on the toolbar.
4. Press Alt-Tab to return to program editing session.
5. Add lines 101, 131 and 161 to the program code as follows:

```
00070 PROCESS_SAL_ID:
00080 IF SAL_ID$="" THEN GOTO *RETURN
00090 REFRESH_FLG=1
00100 READ (SM_FN,KEY=SAL_ID$,DOM=NEW_REC) IOL=1000
00101 GOSUB PROCESS_SAL_COMM
00110 RETURN
00120 NEW_REC:
00130 LET SAL_NAME$="",SAL_COMM$="N",SAL_COMM_RATE=0,SAL_TOTAL=0
00131 GOSUB PROCESS_SAL_COMM
00140 RETURN
00150 !
00160 PROCESS_SAL_COMM:
00161 IF SAL_COMM$="Y" THEN CALL "*wingrp;Enable",COMMISSION.GRP$ ELSE
      :CALL "*wingrp;Disable",COMMISSION.GRP$; LET SAL_COMM_RATE=0
00170 RETURN
```

Save the updated program.

6. Press Alt-Tab to return to the NOMADS designer session.
7. Press the Test button on the tool bar to test the panel and it's attached logic. (During the testing, be sure to create a small sample file which can be used in future labs.)

Lab 1.6: Assigning User Controls

Set a User control to display a message when the F2 key is pressed.

1. Press the 'Ctrls' button on the toolbar to define the User Control.
 - (a) Enter 2 in the Ctl value input field.
 - (b) Select 'Execute' as the process and enter IF SAL_TOTAL>1000000 THEN MSGBOX "Member of the President's Club!","Congratulations!" in the Process input field.
 - (c) Press the 'Add' button.
 - (d) Press 'Done'.
2. Record the changes and test the changes by pressing the appropriate buttons on the toolbar.

Lab 1.7: Creating folders

Create a second panel called SALESTOP with a listbox containing the top salespersons, which can be added, along with the SALESMANT panel, as folders to a third panel called SALES.

1. Create the SALESTOP panel and controls and add LOAD_TOPLIST logic to the SALEPROG program.
 - (a) Quit the SALESMANT panel to return to the Library Object Selection screen. Enter SALESTOP in the 'Name' field and press the 'Panel Object' button.
 - (b) Open the Panel Header definition. In the Display folder, change the title to **President's Club** and set the default program to **SALEPROG**. In the Attributes folder, check Dialogue. In the Logic folder, set the Post-Display logic to Perform ";Load_Toplist".
 - (c) Add fonted text across the top of the panel which reads "More than a million in sales!". Change the font, double it's size, centre it, and make it blue, bold and italics.
 - (d) Add a list box named TOPLIST below the fonted text.
 - (e) Record the changes.
 - (f) Add the following code to SALEPROG and save it:

```
00390 LOAD_TOPLIST:
00400 LET X=HFN; OPEN (X) "SAL_MAST"
00410 REFRESH_FLG=1
00420 LIST_BOX LOAD TOPLIST.CTL, ""
00430 NEXT_KEY: LET K$=KEY(X, END=ALL_DONE)
00440 READ (X, KEY=K$) IOL=1000
00450 IF SAL_TOTAL >= 1000000 THEN LIST_BOX LOAD
      :TOPLIST.CTL, 0, SAL_NAME$
00460 GOTO NEXT_KEY
00470 ALL_DONE:
00480 CLOSE (X)
00490 RETURN
```

2. Create the SALES panel and folders using SALESMANT and SALESTOP.
 - (a) Create a new panel called SALES. Make it 8 columns wider and 8 lines higher than the other two panels.
 - (b) Add folders to the SALES panel. Select the folder control on the tool box and place the control on the panel so that it occupies most of the panel space.
 - (c) In the Tabs/Folder definition, select SALESMANT from the Name drop box and press the 'Insert' button. Highlight <<End>> in the folder list to set the position for the next folder to be inserted. Select SALESTOP from the Name drop box and press 'Insert'. Press 'OK'.
 - (d) Record the changes and test the panel.

Lab 2

Objective

The objective of this lab is to create a GUI application to maintain and display a file using the Data Dictionary.

Description

In this lab, you will create a Data Dictionary for an Account Type file called ACCT_TYP. Also, you will create a new data class PAY_TYPE which will be used by this Data Dictionary. Then you will use the File Maintenance System to automatically generate a maintenance panel called ACCT_MAINT. You will customize the panel by adding help tips. You will also create Query Objects to display data from files both with and without Data Dictionaries.

Lab 2.1: Define a New Data Class

Create a new radio button data class called PAY_TYPE which can be used to select the type of payment (i.e. Cash, Cheque, MasterCard, Visa, Other).

1. RUN “*NOMADS”
2. Create a new data class. Select Dictionary/Classes from the NOMADS II menu.
3. Enter **PAY_TYPE** in the ‘Class Name’ field and answer ‘YES’ to create a new class.
4. Enter **Type of Payment** in the ‘Description’ field. (This will be used as a heading above the buttons in the File Maintenance panel.)
5. Set the size to 1. In this case, size can represent the size of the control or the size of the data. Radio button data is one character long. If size is set to a value greater than one, then the length field must be reset in any Data Dictionary element description which uses this class. Sizes greater than one are used by the file maintenance generator when calculating the size of the radio button control.
6. Select ‘Radio Buttons’.
 - (a) Check the ‘Button on Right’ property.
 - (b) Enter **Cheque** as the ‘Button Text’ , **Q** in the ‘Transtate Value’ box and press the ‘Insert Button’.
 - (c) Enter **Cash** as the ‘Button Text’ , **C** in the ‘Translate Value’ box and press the ‘Insert Button’.
 - (d) Enter **MasterCard** as the ‘Button Text’ , **M** in the ‘Translate Value’ box and press the ‘Insert Button’.
 - (e) Enter **Visa** as the ‘Button Text’ , **V** in the ‘Translate Value’ box and press the ‘Insert Button’.
 - (f) Enter **Other** as the ‘Button Text’ , **O** in the ‘Translate Value’ box and press the ‘Insert Button’.
 - (g) Highlight **Cheque** in the radio button list and press ‘Set Default’. This sets the default value for the radio button set.

7. Press 'Write' and PAY_TYPE is added to the data class file.
8. Press 'Exit'.

Lab 2.2: Create a Data Dictionary

Create a Data Dictionary for the Account Type file called ACCT_TYP. It consists of four data fields which are:

AT_ID (Key)	Account type code – a unique four character account type identifier
AT_TERMS	Two character terms code to identify the applicable terms
AT_DISCOUNT	Numeric value containing the percent discount
AT_PAY_TYPE	A single character code indicating the mode of payment: Q – Cheque C – Cash M – MasterCard V – Visa O – Other

There are two keys associated with the file:

Primary key	AT_ID
Alternate key 1	AT_PAY_TYPE + AT_TERMS

The primary key is not external.

1. Invoke Data Dictionary Maintenance. Select Dictionary/Maintenance from the NOMADS II menu.
2. Enter **Account Type** as the Logical File name and **ACCT_TYP** as the Physical File name. Answer 'YES' to create the new definition.
3. Define the Data Elements. Click on the Data Elements list box and press 'Insert Element' to begin element definition. Enter the definition for each element, press 'OK', then select 'Insert Element' again to add the next. Define the data elements as follows:

Name:	AT_ID
Required:	Check this.
Short Desc:	Account Type Code
Type:	String
Length:	4
In the Query Tab:	
Library:	WRKLIB.EN
Panel:	ACCT_QRY
In the User Aids Tab:	
Longwinded desc:	A unique four character account identifier.

Name:	AT_TERMS
Required:	Check this.
Short Desc:	Terms Code
Type:	String
Length:	2
In the User Aids Tab:	
Longwinded desc:	Two character terms code to identify the applicable terms.

Name: AT_DISCOUNT
Short Desc: Discount (%)
Type: Numeric
Length: 5
Print Fomat #0.00
In the User Aids Tab:
Longwinded desc: Percent Discount

Name: AT_PAY_TYPE
Class: PAY_TYPE
Required: Check this.
Short Desc: Type of Payment
Type: String
Length: 1
Default value: Q
In the User Aids Tab:
Longwinded desc: Mode of Payment indicator

4. Define the keys. Press the 'Define Keys' button.
 - (a) Primary key definition: Highlight AT_ID in the Data Fields list and press 'Add Ascending' to add it to the Key Segments list. Press 'OK'.
 - (b) Alternate Key 1: Press 'Add key'. Select AT_PAY_TYPE from the Data Fields and press 'Add Ascending'. Then double click AT_TERMS in the Data Fields list to add it to the Key Segments list as well. Press 'OK'.
 - (c) Exit the Key Definition panel.
5. Create the file with the embedded iolist. Press 'Update Physical'.
6. Exit Data Dictionary Maintenance.

Lab 2.3: Generate a File Maintenance Panel

Generate a File Maintenance panel called ACCT_MAINT based on the Account Type data dictionary definition. It will look like this:

The screenshot shows a window titled "File Maintenance for Account Type". It contains the following fields and controls:

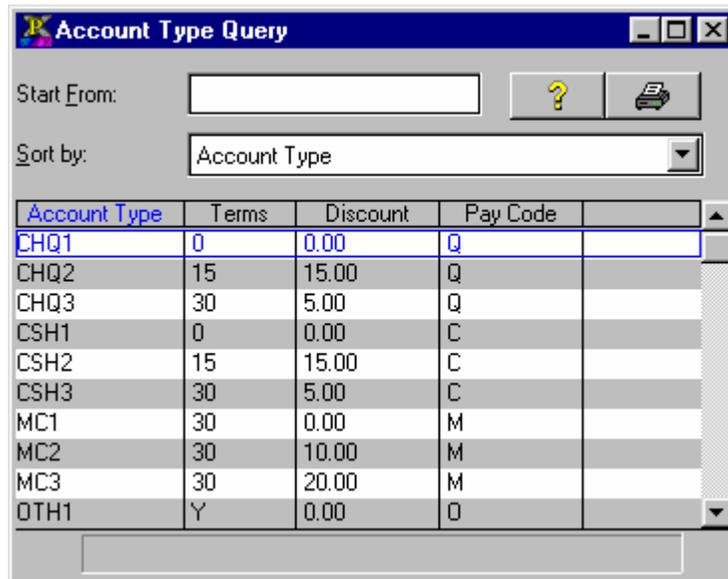
- Account Type Code: ?
- Terms Code:
- Discount(%):
- Type of Payment:
 - Cheque:
 - Cash:
 - MasterCard:
 - VISA:
 - Other:
- Toolbar: Write, Delete, navigation arrows, Clear, Exit

1. Open the **wrklib.en** library. Select Library/Open from the NOMADS II menu, and open the **wrklib.en** library.
2. Invoke the File Maintenance system. Enter **ACCT_MAINT** in the 'Name' field of the Library Object Selection screen and press the 'File Maint' button.
3. In the Parameter Selection folder, select **Account Type** from the Data Dictionary drop box. Select the 'Auto clear' New Records option. In Message Options, remove the checks from 'Confirm New Record' and 'Acknowledge Deletes'.
4. Press the 'Generate' button.
5. Customize the ACCT_MAINT panel. Select the ACCT_MAINT panel from the Library Objects list of the Library Objects Selection screen. Re-align controls if required. Then open the AT_ID multiline definition and check the Uppercase attribute in the Attribute folder. Press 'OK'. Record the changes.
6. Press the 'Test' button on the Library Object Selection screen to test the panel. Add some records to create a small sample ACCT_TYP file.

Lab 2.4: Create a Query Using the Data Dictionary

Create a query called ACCT_QRY to display the data in the Account Type file and return the key of the selected record.

ACCT_QRY will look like this:



Account Type	Terms	Discount	Pay Code
CHQ1	0	0.00	Q
CHQ2	15	15.00	Q
CHQ3	30	5.00	Q
CSH1	0	0.00	C
CSH2	15	15.00	C
CSH3	30	5.00	C
MC1	30	0.00	M
MC2	30	10.00	M
MC3	30	20.00	M
OTH1	Y	0.00	O

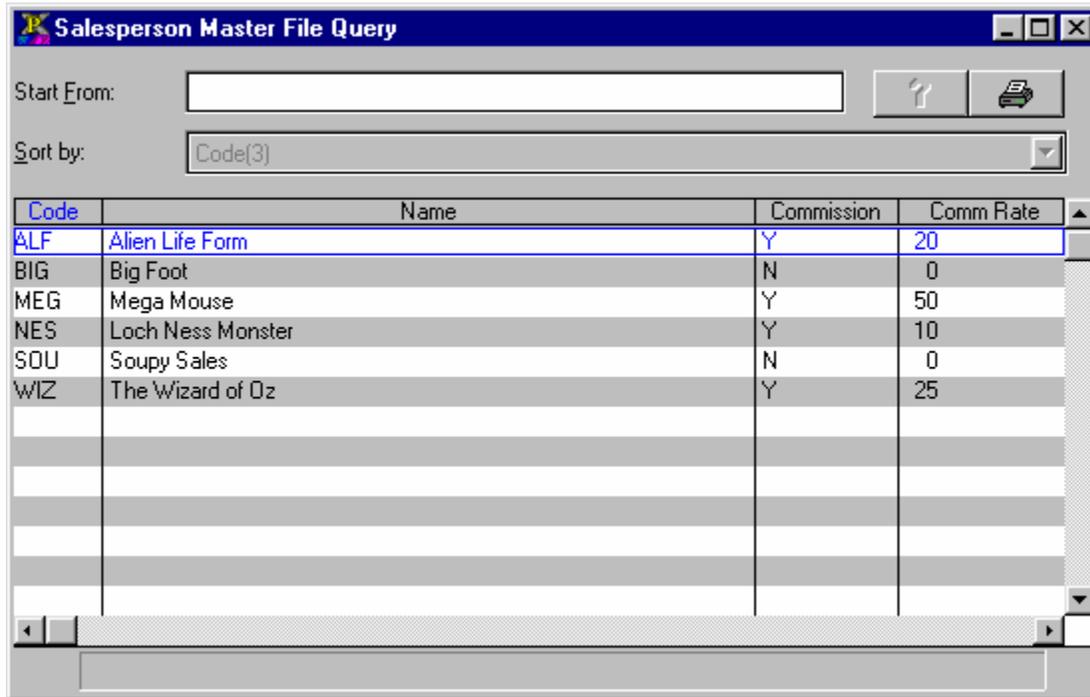
1. Create a Query object. Enter **ACCT_QRY** in the 'Name' field of the Library Object Selection screen and press the 'Query object' button.
2. Define the Query Header.
 - (a) In the File Information folder of the Query Header Definition, select **Account Type** from the 'File' drop box. You may enter **AT_ID\$** in the 'Return value' input field or leave it blank, as the default return value when left blank is the value in the first column, which in this case is AT_ID.
 - (b) In the Display folder, set the position to 'Centered' and set the size to a width of 45 and a height of 16. Change the 'Title' to **Account Type Query**.
 - (c) Press 'OK'.
3. Select the file elements to be displayed in the query.
 - (a) Highlight AT_ID in the File Elements list and press 'Add column'.
 - (b) Double click the mouse on AT_DISCOUNT and AT_PAY_TYPE to add these fields to the Columns list.
 - (c) Highlight AT_DISCOUNT in the Columns list and double click AT_TERMS in the File Elements list. This will insert AT_TERMS before AT_DISCOUNT.

4. Adjust the display options for each column.
 - (a) Select AT_ID in the Columns list by double clicking the mouse on it. Change the title to **Account Type** and the width to 10. (Note that AT_ID uses upper case letters, which, in the case of some proportional fonts, require extra space to be displayed. So even though the code is four characters wide, set the display width to a larger value.) Press 'OK'.
 - (b) Select AT_TERMS in the Columns list. Change the title to **Terms** and the width to 6.
 - (c) Select AT_DISCOUNT in the Columns list. Change the title to **Discount**, format mask to **#0.00%** and width to 8.
 - (d) Select AT_PAY_TYPE and change the title to **Pay Code** and the width to 8.
5. Press the 'Test' button to test the query. Select a record either by double clicking the mouse on it or by highlighting it and pressing the **<Enter>** key. The value from the first column is returned.
6. Exit the query definition.
7. Highlight ACCT_MAINT in the Library Objects list and press the 'Test' button. Now press the '?' button next to the Account Type Code multiline. The query will be displayed and the return value is placed in the multiline associated with the query.

Lab 2.5: Create a Manual Query

Create a manual query called SALESQRY to display the data in the SAL_MAST file and return the key of the selected record.

SALESQRY will look like this:



Code	Name	Commission	Comm Rate
ALF	Alien Life Form	Y	20
BIG	Big Foot	N	0
MEG	Mega Mouse	Y	50
NES	Loch Ness Monster	Y	10
SOU	Soupy Sales	N	0
WIZ	The Wizard of Oz	Y	25

1. Create a Query object. Enter **SALESQRY** in the 'Name' field of the Library Object Selection screen and press the 'Query object' button.
2. Define the Query Header.
 - (a) In the File Information folder of the Query Header Definition, enter SAL_MAST into the 'File' field. Confirm that SAL_MAST is not defined in the Data Dictionary and must be defined manually.
 - (b) In the Display folder, set the position to 'Centred' and set the size to a width of 68 and a height of 20. Change the 'Title' to **Salesperson Master File Query** and set 'Static Columns' to 2.
 - (c) Press 'OK'.

3. Define the file elements to be displayed in the query.

- (a) Define the key field. Press 'Add Column'. The key is external and is identified as field number zero. Enter **0** in 'Field No', **Code** in 'Title' and **5** in 'Width'. Press 'OK'
- (b) Define the name field. Press 'Add Column'. The name field is the first field in the data iolist, so enter **1** in 'Field No'. Set 'Title' to **Name** and 'Width' to **40**.
- (c) Add the rest of the fields:

Field No	Type	Title	Format	Width
2	String	Commission		8
3	Numeric	Comm Rate	#0	8
4	Numeric	Sales to Date	####,###,##0.00	15

- 4. Press the 'Test' button to test the query. Exit the query definition.
- 5. Select the SALESMAINT panel in the Library Objects list. Open the SAL_ID multiline definition and select the Query folder. Enter SALESQRY in the 'Panel' input field. Press 'OK'. Record the change and exit the panel definition.
- 6. Highlight SALESMAINT in the Library Objects list and press the 'Test' button. Now press the query button next to the Salesperson ID multiline. Select a record from the query to test the return value.

Lab 2.6: Attaching Help Using the Status Bar

Attach help to ACCT_MAINT by displaying the longwinded descriptions from the Data Dictionary in the status bar.

1. Add the status bar to the ACCT_MAINT panel. Select the ACCT_MAINT panel from the Library Objects list of the Library Objects Selection screen. Open the Header definition and check Status Bar in the Attributes folder. Press 'OK'. Record the changes. The longwinded descriptions will automatically display in the status bar when ACCT_MAINT is processed.

Lab 2.7: Adding Tips

Attach help to ACCT_MAINT by displaying Floating Tips for each of the controls.

1. Add the floating tips to the controls on the ACCT_MAINT panel. Select the ACCT_MAINT panel from the Library Objects list of the Library Objects Selection screen.
 - (a) Open the AT_ID control definition and enter an appropriate tip in the 'Floating Tip' field of the User Aids folder. Press 'OK'.
 - (b) Repeat step (a) for the following controls:

AT_TERMS	
AT_DISCOUNT	
AT_PAY_TYPE:Cheque:	
AT_PAY_TYPE:Cash:	
AT_PAY_TYPE:MasterCard:	
AT_PAY_TYPE:Visa:	
AT_PAY_TYPE:Other:	
BUTTON_1	(Write button)
BUTTON_2	(Delete button)
BUTTON_3	(Exit button)
BUTTON_4	(First record button)
BUTTON_5	(Prior record button)
BUTTON_6	(Next record button)
BUTTON_7	(Last record button)
BUTTON_8	(Clear record button)

2. Record the changes and test.

Lab 3

Objectives

The objectives of this lab are:

- To create a more complex file maintenance and query application which makes use of cross-reference files.
- To convert an existing program to multi-lingual capability.
- To create a menu system to launch an integrated application.

Description

In this lab you will create a Data Dictionary for a Client Master file called CL_MAST, and generate an associated File Maintenance panel called CL_MAINT with two associated folder panels (CL_MAINT.1 and CL_MAINT.2) and a custom program called CL_UPDT. You will reference a cross-reference file (SAL_MAST) using a Validator and Formatter and build a query called CL_QRY which also uses link files. Then you will create a new message library called **wrklib.msg** and convert SALEPROG to retrieve messages from it. Also, You will build the MENU panel to integrate and launch the Salesperson Information, Account Type Maintenance and Client Master applications.

Lab 3.1: Create a Data Dictionary

Create a Data Dictionary for the Client Master file called CL_MAST. It consists of a primary key and four data fields which are:

CL_ID (Key)	Client ID – a unique six character client identifier
CL_NAME	Name of the client
CL_PHONE	Client's phone number
CL_SMN	Three character code for the salesperson responsible for the client
CL_TYPE	Four character code indicating the account type

There are three keys associated with the file:

Primary key	CL_ID
Alternate key 1	CL_NAME + CL_ID
Alternate key 2	CL_TYPE + CL_ID

The primary key is external and does not form part of the data iolist.

1. RUN “*NOMADS”
2. Invoke Data Dictionary Maintenance. Select Dictionary/Maintenance from the NOMADS II menu.
3. Enter **Client Master** as the Logical File name and **CL_MAST** as the Physical File name. Answer ‘YES’ to create the new definition.

4. Define the following Data Elements:

Name: CL_ID
External only: Check this. (This means the field is not included in the data iolist)
Required: Check this.
Short Desc: Client ID
Type: String
Length: 6
In the Query Tab:
Library: WRKLIB.EN
Panel: CL_QRY
In the User Aids Tab:
Longwinded desc: Client ID – A unique six character client identifier.

Name: CL_NAME
Required: Check this.
Short Desc: Name
Type: String
Length: 40
In the User Aids Tab:
Longwinded desc: Name of the client.

Name: CL_PHONE
Short Desc: Phone
Type: String
Length: 11
Print format: 000-000-0000
In the User Aids Tab:
Longwinded desc: Phone number

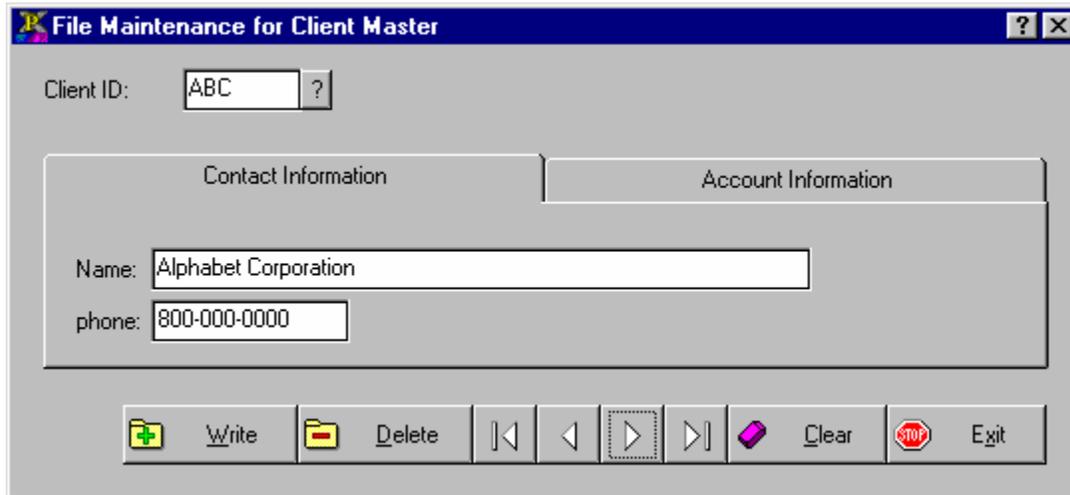
Name: CL_SMN
Short Desc: Salesperson Code
Type: String
Length: 3
In the Query Tab:
Library: WRKLIB.EN
Panel: SALESQRY
In the User Aids Tab:
Longwinded desc: Three character code for the salesperson responsible for the client.

Name: CL_TYPE
Required: Check this.
Short Desc: Account Type
Type: String
Length: 4
In the Query Tab:
Library: WRKLIB.EN
Panel: ACCT_QRY
In the User Aids Tab:
Longwinded desc: Four character code indicating the account type.

5. Define the keys. Press the 'Define Keys' button.
 - (d) Primary key definition: Check External Key. Then highlight CL_ID in the Data Fields list and press 'Add Ascending' to add it to the Key Segments list. Press 'OK'.
 - (e) Alternate Key 1: Press 'Add key'. Select CL_NAME from the Data Fields and press 'Add Ascending'. Then double click CL_ID in the Data Fields list to add it to the Key Segments list as well. Press 'OK'.
 - (f) Alternate Key 2: Press 'Add key'. Select CL_TYPE and CL_ID to form the Key Segments. Press 'OK'.
 - (g) Exit the Key Definition panel.
6. Create the file with the embedded iolist. Press 'Update Physical'.
7. Exit Data Dictionary Maintenance.

Lab 3.2: Generate a Custom File Maintenance Panel

Generate a custom File Maintenance panel called CL_MAINT with two associated folder panels (CL_MAINT.1 and CL_MAINT.2) and program called CL_UPDT based on the Client Master data dictionary definition. The panel will look like this:



1. Open the **wrklib.en** library. Select Library/Open from the NOMADS II menu, and open the **wrklib.en** library.
2. Invoke the File Maintenance system. Enter **CL_MAINT** in the 'Name' field of the Library Object Selection screen and press the 'File Maint' button.
3. In the Parameter Selection folder, select **Client Master** from the 'Data Dictionary' drop box. Enter **CL_UPDT** in the 'Maintenance Program' field and answer 'YES' to use the pathname even though it does not yet exist. Check 'Custom' as the Program Type option. Then select 'Folders' in Display Options, and answer 'YES' to move the data fields off the main panel. Select the 'Auto clear' New Records option. In Message Options, remove the checks from 'Confirm New Record' and 'Acknowledge Deletes'.
4. Select the 'Folder Definition' folder.
 - (a) Enter **Contact Information** in 'Folder Tab Label' and press 'Insert Folder' button.
 - (b) Enter **Account Information** in 'Folder Tab Label' and press 'Insert Folder' button.
5. Select the 'Field Selection' folder.
 - (a) Select **Contact Information** from the 'Folder' drop box. Select CL_NAME from the Unselected items list and press 'Include Item' to add it to the Selected items. Double click on CL_PHONE to add it to the selected items.
 - (b) Select **Account Information** from the 'Folder' drop box. Add CL_SMN and CL_TYPE to the list of selected items.
6. Press the 'Generate' button.

7. Customize the CL_MAINT.2 panel by selecting it from the Object list. Then open the CL_SMN and CL_TYPE controls and check the Uppercase attribute in the Attributes folder. Press 'OK'. Record the changes.
8. Select CL_MAINT from the Object List and test the panel. Add some records, being sure to use the same salesperson codes you used in the SAL_MAST data file and account type codes you set up in the ACCT_TYP data file.

Lab 3.3: Validation/Formatting Using Cross-Reference Files

Customize the File Maintenance program CL_UPDT to pull data from the secondary file. The primary file is CL_MAST and the secondary file is SAL_MAST.

1. Open the **wrklib.en** library. Select *Library/Open* from the NOMADS II menu, and open the **wrklib.en** library.
2. Customize the CL_MAINT.2 panel. Select it from the Library Objects list and open the CL_SMN control. In the Validation folder, enter **CL_UPDT;Process_Salesman** in the 'Validator' input field and **CL_UPDT;Process_Salesman** in the 'Formatter' input field. In the Logic folder, for the On Select Logic select 'Execute' from the drop box and enter **REFRESH_FLG=1** in the input field. Press 'OK'. Record the changes.
3. Add a new multiline to the CL_MAINT.2 panel to display the salesman name from the SAL_NAME field in the cross-reference file SAL_MAST. Place it next to the CL_SMN (Salesperson Code) multiline. Call it **%SAL_NAME** and check the 'Borderless' and 'Locked' attributes in the Attributes folder, and remove the check from 'Tab Stop'. Record the changes and exit the panel definition.
4. Start a new PVX session and load the CL_UPDT program.
5. Add the following code to CL_UPDT.

```
20000 ! 20000
20010 PROCESS_SALESMAN:
20020 ENTER IN_DATA$,ERR_MSG$,ERR=*NEXT
20030 LET %SAL_NAME$=""; IF IN_DATA$="" THEN GOTO DONE
20040 LET X=FFN("SAL_MAST")
20050 IF X=-1 THEN LET XX=HFN; OPEN (XX)"SAL_MAST"; LET X=XX
20060 READ (X,KEY=IN_DATA$,DOM=NOT_THERE) SAL_NAME$,SAL_COMM$,
      :SAL_COMM_RATE,SAL_TOTAL
20070 LET %SAL_NAME$=SAL_NAME$
20080 GOTO DONE
20090 NOT_THERE:
20100 LET ERR_MSG$="Salesman does not exist"
22020 DONE:
22030 IF XX THEN CLOSE (XX)
22040 EXIT
```

Save the program.

6. Press Alt-Tab to return to the NOMADS session.
7. Select CL_MAINT from the Library Objects list. Press the 'Test' button to test the panel. Select a record from the primary file. Tab to the CL_SMN input field and test the validation/formatter logic.

4. Create the link to the SAL_MAST file.
 - (a) Enter **SAL_MAST** in the 'File' input field. Press the 'OK' button to confirm that SAL_MAST is not in the Data Dictionary and must be defined manually.
 - (b) Enter **CL_SMN\$** in 'Key definition'.
 - (c) Press 'Write'.
 - (d) Press 'Exit'.

5. Select the elements to be displayed.
 - (a) Add the following elements to the Columns list:
 - CL_ID
 - CL_NAME
 - CL_PHONE
 - CL_SMN
 - CL_TYPE
 - ACC.AT_TERMS
 - ACC.AT_DISCOUNT
 - ACC.AT_PAY_TYPE

 - (b) Adjust the display options for the following columns:

Element	Change title...	Change format...	Change width...
CL_NAME			25
CL_SMN	Smn		
CL_TYPE	Acct Cd		7
ACC.AT_TERMS	Terms		5
ACC.AT_DISCOUNT	Discount	#0.00%	8
ACC.AT_PAY_TYPE	Pay		3

 - (c) Insert the salesperson name field from the manually-defined file. Highlight the *File=SAL_MAST entry in the File Elements list, then highlight CL_TYPE in the Columns list. Press 'Add column' to insert a field from the manually-defined file. Enter 1 in 'Field No', Salesperson in 'Title' and 25 in 'Width'. Press 'OK'.

6. Test the query.

7. Exit the query definition.

8. Test the CL_MAINT panel. Press the query button next to the Client ID input field and select a record.

Lab 3.5: Using a Message Library

Create a message library to be used by the wrklib.en library. Change the SALEPROG program and update the buttons on the SALESMAINT panel to access the messages.

1. Create a new message library called **wrklib.msg** for the **wrklib.en** library.
 - (a) From the Library Object Selection panel, press the 'Library Defaults' button.
 - (b) Enter **wrklib.msg** in the 'Message Lib' input field. Answer 'YES' to create the file.
 - (c) Press 'OK'.
2. Add the messages to the **wrklib.msg** file.
 - (a) Exit the Library Object Selection panel. Select *Options/Message Manager* from the NOMADS II menu.
 - (b) Enter **wrklib.msg** in the 'Library File' input field.
 - (c) Enter **MISS_SMN** in the 'Message Key' field and **Missing Salesperson Code** in the 'Text' field. Press 'Write'.
 - (d) Repeat the previous step to add the following messages:

<u>Message Key</u>	<u>Text</u>
ERROR	Error
REC_UPDT	Record %1 updated
UPDATE	Update
REC_DEL	Record %1 deleted
DELETE	Delete
REC_NOEXST	The record %1 does not exist.
&WRITE	&Write
&DELETE	&Delete
E&XIT	E&xit

- (e) Exit Message Library Maintenance.
3. Change SALEPROG to access the messages.
 - (a) Start a new session of PVX and load the SALEPROG program.
 - (b) Change line 200 from:

```
00200 IF SAL_ID$="" THEN MSGBOX "Missing Salesperson Code",
      : "Error"; LET NEXT_ID=SAL_ID.CTL; GOTO *RETURN
```

to:

```
00200 IF SAL_ID$="" THEN MSGBOX MSG("MISS_SMN"), MSG("ERROR");
      : LET NEXT_ID=SAL_ID.CTL; GOTO *RETURN
```
 - (c) Change line 220:

```
00220 MSGBOX MSG("REC_UPDT", SAL_ID$), MSG("UPDATE")
```
 - (d) Change line 280:

```
00280 MSGBOX MSG("REC_DEL", SAL_ID$), MSG("DELETE")
```
 - (e) Change line 320:

```
00320 MSGBOX MSG("REC_NOEXST", SAL_ID$), MSG("DELETE")
```
 - (f) Save the program.

4. Change the buttons on the SALESMAINT panel to use the message library.
 - (a) Alt-Tab back to the NOMADS session. Select *Library* from the menu and open the **wrklib.en** library.
 - (b) Select the SALESMAINT panel from the Library Objects list.
 - (c) Open the WRITE button control. In the Display folder, change the 'Text' option to Expression and enter "{!file_add}"+msg("&WRITE") in the 'Text' input field. Press 'OK'.
 - (d) Open the DELETE button control. In the Display folder, change the 'Text' option to Expression and enter "{!file_del}"+msg("&DELETE") in the 'Text' input field. Press 'OK'.
 - (e) Open the EXIT button control. In the Display folder, change the 'Text' option to Expression and enter "{!X}"+msg("&EXIT") in the 'Text' input field. Press 'OK'.
 - (f) Record the changes and exit the panel.
5. Test the messages. Highlight SALESMAINT in the Library Objects list and press 'Test'. Add and delete records to display the messages.

Lab 3.6: Building a Menu

Create a menu to integrate and launch the Salesperson, Account Type and Client applications:



1. Create a panel called MENU.
 - (a) Enter MENU in the 'Name' input field of the Library Object Selection screen and press the 'Panel Object' button.
 - (b) Define the Panel Header. In the Display folder, set 'Width' to 70 and 'Height' to 16. In the Attributes folder, check 'Dialogue', 'Menu Bar' and 'Suppress.VAL'. Press 'OK'.
 - (c) Record the changes.
2. Create a button named EXIT on the bottom right corner of the panel. Enter **E&xit** in the 'Text' input field and select a bitmap (such as !X). Set 'Column' to 52, 'Line' to 13, 'Width' to 15 and 'Height' to 2. In the Logic folder, select END from the 'When Button Pressed' drop box. Press 'OK'.
3. Assign a bitmap as the background for the panel. Place an Image control on the panel. In the Display folder, enter an appropriate bitmap path (such as C:\windows\clouds.bmp) in the 'Picture Path' input field. Set 'Column' to 0, 'Line' to 0, 'Width' to 70 and 'Height' to 16. Press 'OK'.
4. Create the menu bar.
 - (a) Enter the Menu Bar Maintenance Utility. Select **Panel/Menus** form the Panel definition menu bar.
 - (b) Press the 'New Item' button. Enter **&Salesperson Information** in the 'Item' input field. Select 'Link' from the 'Function' drop box and enter "**SALESMAINT**" in the associated input field. Press 'OK'.

(c) Repeat step (b) to add the following menu items:

<u>Item</u>	<u>Function</u>
&Account Type Maintenance	Link "ACCT_MAINT"
&Client Master	Link "CL_MAINT"

(d) Press 'OK' to create the menu records. Record the changes.

(e) Press the 'Test' button on the Library Object Selection screen to test the panel.

Lab 3.7: Creating a New Shortcut

Create an Icon on your desktop to run the menu.

1. In Windows, create a shortcut.

(a) Set the 'Target' to the path of the PVXWIN executable with the applicable arguments.

Example:
C:\PVX\PVXWIN.EXE *winstar -arg MENU wrklib.en

(b) Enter the your current directory in 'Start in Directory'.