Building Smartphone Applications

PxPlus 2019 (v16)

DireXions 2019

Goal of this Presentation

To provide guidance on the development of PxPlus Web based applications targeting Smartphones.

Agenda

- Review of sample application
- Understanding design criteria
    - Dealing with the physical dimensions of device
    - User input
- Approaches taken
- Application logic
- Stylizing the application

Sample Application

- Previewed in mailing leading up to DireXions

- Basic CRM style application
  - Designed for use by a sales rep in the field
  - Provides access to:

    - Clients - General info, location, invoices and sales history
    - Product - General info, images, invoices and sales history
    - Invoices - General info, products ordered, status and delivery details
    - Sales - Analysis of past sales and trends

Designing for Smartphones

Primary Consideration

- Similar to real estate where the Mantra is

  Location, Location, Location !!

- For smartphones the Mantra is

  Real Estate, Real Estate, Real Estate !!

Designing for Smartphones

Primary Consideration

- Characters must be visible
  - Good screen sizes
    - Portrait 40-45 characters wide | 40-42 lines high
    - Landscape 70-80 characters wide | 20-24 lines high
  - Smartphones tend to scale down oversize forms
    - Avoid scaling if possible
  - Use resizable screens to allow system to maximize space
    - Enable auto-maximize on your *i*Nomads template
    - Design screen smaller than smallest phone you want to support

Designing for Smartphones

Clean User Interface

- Simplify user input
  - For navigation, use “something” to click or touch
    - Buttons, menus, direct links, etc.
  - Wherever possible, use lists to select items
    - Lists can be scrollable or paged – but be consistent
      - Scrollable is more user friendly with smartphones
        - *i*Nomads provides load on display for long lists automatically

Keyboard input is burdensome

DireXions 2019

APPLICATION WALK THRU

What is the Goal?

Application Walk Through

- Reasons to walk through the application
  - Provide ideas on how to address smartphone applications
  - Demonstrate capabilities of PxPlus and Nomads/*i*Nomads toolset
  - Help you understand how to modify code to suit your needs
  - Show some CSS tweaks to improve look of application

The Basics

- Demo found in *demos/salesrep

  - Panel Library - scrnlib.en
  - Data Files - client, products, invhdr, invdtl, salesrep, department, states

  - Images - images/products/*, images/salesrep/*, images/logo.png
  - Dictionary - providex.*
  - Views - pvxview.*
  - Reports - reports/*
  - Query Info - query.inf

The Basics

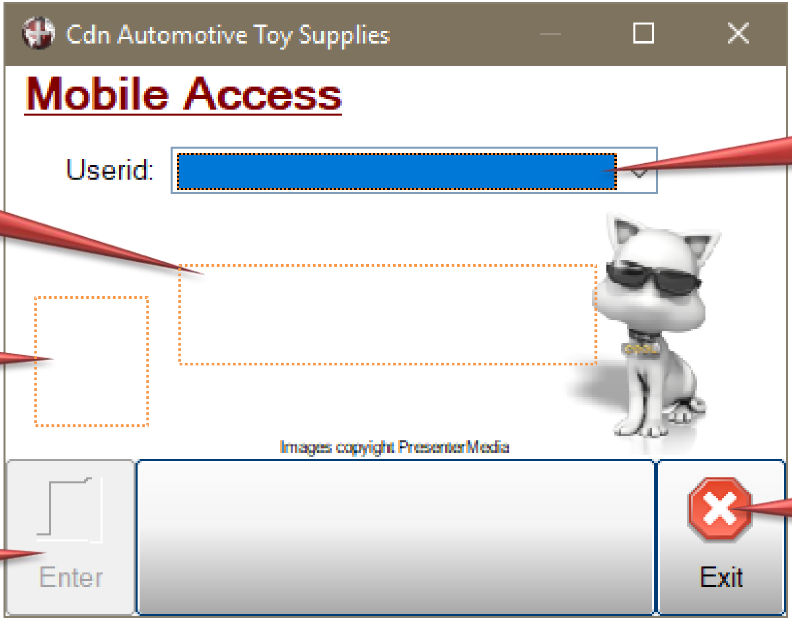
- Demo found in *demos/salesrep *(continued)*
  - Programs

Program/Object	Description
custinfo.pvc	Object used to display customer information
prodinfo.pvc	Object used to display product information
invinfo.pvc	Object used to display invoice information
repinfo.pvc	Object used to display sales rep information
phoneapp.pvc	Object module with common functions
chrt_*	Chart generation programs

Logon Panel

- End-user identification
  - For security purposes, typically require some user identification
    - eg. email, user code, etc.
    - Demo simply select a sales rep

Logon Panel



Dummy list box used to enable resizing

Sales rep photo in dynamic image control

Button to logon *(enabled once rep selected)*

Drop box populated with sales reps

Button to exit application

Logon Panel

- Logic
  - Sales rep drop box (*repname*) loaded using SmartLoad of "QSalesRep"
  - On select, extracts sales rep code, reads sales rep file, and sets photo pathname

```
if repname$<>""
    salesrep$=mid(repname$, -3, 2);
    read ("salesrep",  key=salesrep$,  rec=%rep$, dom=*next);
    photo$=pth("images/salesrep/"+lcs(salesrep$)+".png",  "images/salesrep/noimage.png")
```

Direct File IO without Open/Close

Use multiple pathnames to specify alternate

  - Dependency table enables "Enter" button and shows photo
  - Enter button sets Replacement_Scrn$ to "Main"

Use Dependency tables to control screen

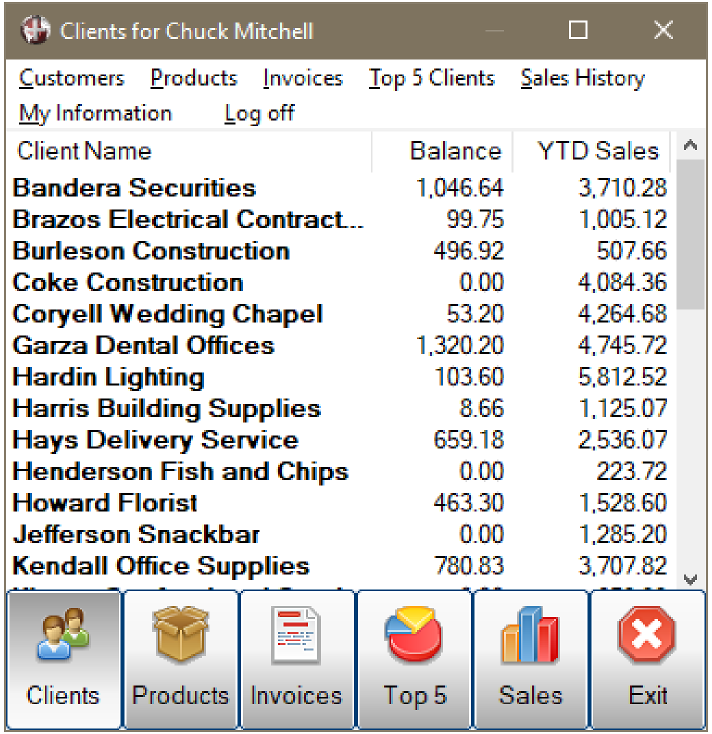
Replace current screen as opposed to new window

Main Panel

- Main selector for the application
  - Contains buttons to select display
    - List of clients, products, or invoices
    - Top five clients for sales rep
    - Historical sales history

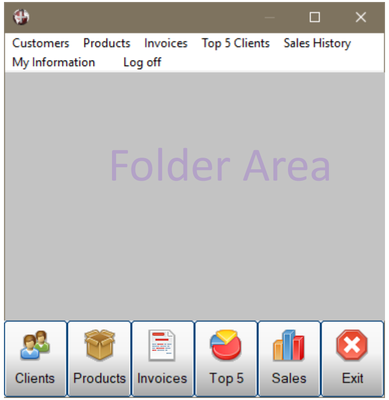
… and an Exit button

Data Selector/Viewer

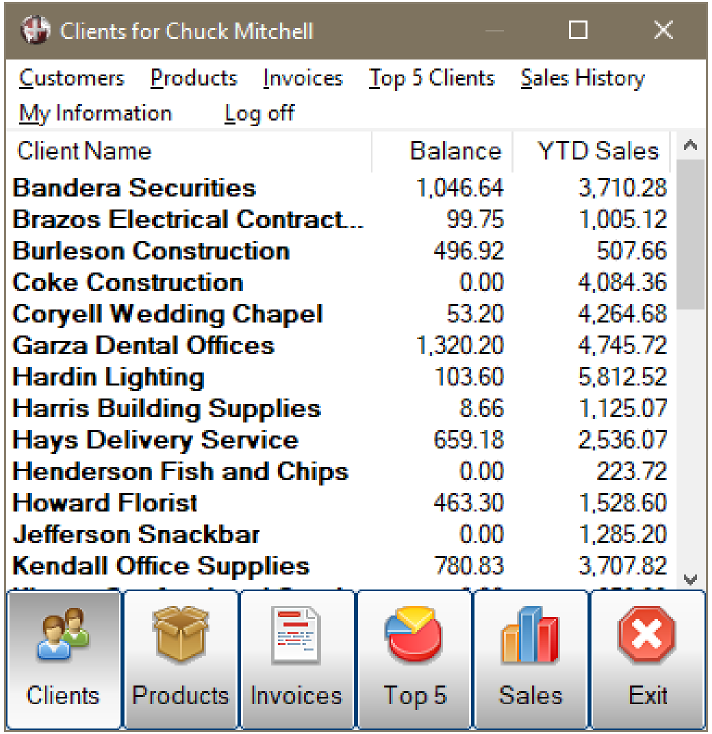


Client Name	Balance	YTD Sales
Bandera Securities	1,046.64	3,710.28
Brazos Electrical Contract...	99.75	1,005.12
Burleson Construction	496.92	507.66
Coke Construction	0.00	4,084.36
Coryell Wedding Chapel	53.20	4,264.68
Garza Dental Offices	1,320.20	4,745.72
Hardin Lighting	103.60	5,812.52
Harris Building Supplies	8.66	1,125.07
Hays Delivery Service	659.18	2,536.07
Henderson Fish and Chips	0.00	223.72
Howard Florist	463.30	1,528.60
Jefferson Snackbar	0.00	1,285.20
Kendall Office Supplies	780.83	3,707.82

Main Panel



Data Selector/Viewer

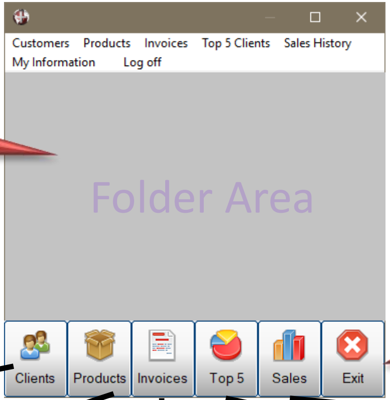


Client Name	Balance	YTD Sales
Bandera Securities	1,046.64	3,710.28
Brazos Electrical Contract...	99.75	1,005.12
Burleson Construction	496.92	507.66
Coke Construction	0.00	4,084.36
Coryell Wedding Chapel	53.20	4,264.68
Garza Dental Offices	1,320.20	4,745.72
Hardin Lighting	103.60	5,812.52
Harris Building Supplies	8.66	1,125.07
Hays Delivery Service	659.18	2,536.07
Henderson Fish and Chips	0.00	223.72
Howard Florist	463.30	1,528.60
Jefferson Snackbar	0.00	1,285.20
Kendall Office Supplies	780.83	3,707.82

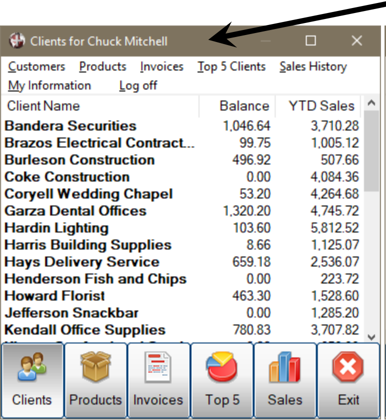
Main Panel

Data Selector/Viewer

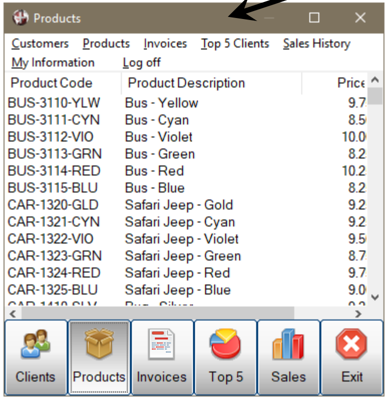
"Tabless" Folder used to display information



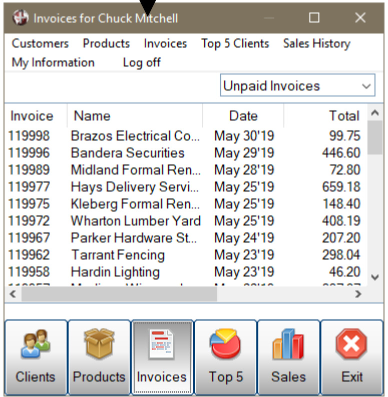
Radio Buttons select the folder



Client Name	Balance	YTD Sales
Bandera Securities	1,046.64	3,710.28
Brazos Electrical Contract...	99.75	1,005.12
Burleson Construction	496.92	507.66
Coke Construction	0.00	4,084.36
Coryell Wedding Chapel	53.20	4,264.68
Garza Dental Offices	1,320.20	4,745.72
Hardin Lighting	103.60	5,812.52
Harris Building Supplies	8.66	1,125.07
Hays Delivery Service	659.18	2,536.07
Henderson Fish and Chips	0.00	223.72
Howard Florist	463.30	1,528.60
Jefferson Snackbar	0.00	1,285.20
Kendall Office Supplies	780.83	3,707.82

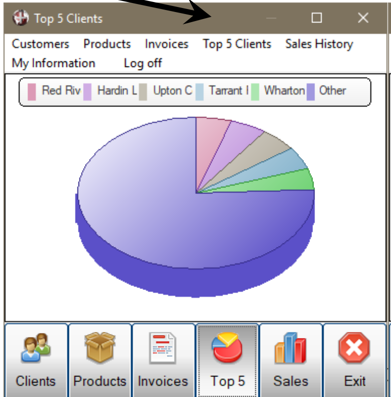


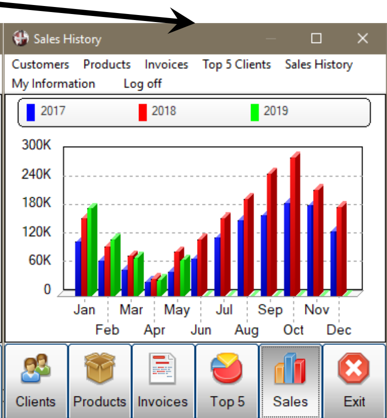
Product Code	Product Description	Price
BUS-3110-YLW	Bus - Yellow	9.7
BUS-3111-CYN	Bus - Cyan	8.5
BUS-3112-VIO	Bus - Violet	10.0
BUS-3113-GRN	Bus - Green	8.2
BUS-3114-RED	Bus - Red	10.2
BUS-3115-BLU	Bus - Blue	8.2
CAR-1320-GLD	Safari Jeep - Gold	9.2
CAR-1321-CYN	Safari Jeep - Cyan	9.2
CAR-1322-VIO	Safari Jeep - Violet	9.5
CAR-1323-GRN	Safari Jeep - Green	8.7
CAR-1324-RED	Safari Jeep - Red	9.7
CAR-1325-BLU	Safari Jeep - Blue	9.0



Unpaid Invoices

Invoice	Name	Date	Total
119998	Brazos Electrical Co...	May 30'19	99.75
119996	Bandera Securities	May 29'19	446.60
119989	Midland Formal Ren...	May 28'19	72.80
119977	Hays Delivery Servi...	May 25'19	659.18
119975	Kleberg Formal Ren...	May 25'19	148.40
119972	Wharton Lumber Yard	May 25'19	408.19
119967	Parker Hardware St...	May 24'19	207.20
119962	Tarrant Fencing	May 23'19	298.04
119958	Hardin Lighting	May 23'19	46.20





Main Panel

Folder Selection

- Folder selection done by BTN_BAR
  - Radio button with six buttons across bottom of screen
    - Buttons have values 1-5 and 9
  - Execute logic with BTN_BAR *(on first button)*

```
btn_bar=num(btn_bar$);
if btn_bar=9
   then  cmd_str$="E"
   else next_folder=11000+btn_bar
```

Folders have fixed CTL/ID values starting at 11000

Main Panel

Menu Logic

- Panel Menu for direct selection of display

Menu Option	Associated Logic
&Customer	Set Next_folder = 11001
&Products	Set Next_folder = 11002
&Invoices	Set Next_folder = 11003
&Top 5 Clients	Set Next_folder = 11004
&Sales History	Set Next_folder = 11005
&My Information	Process panel "RepInfo" (via LINK command)
&Log Off	End panel

Main Panel

Folder Assignments

- "Tabless" folder tabs/panels

Tab	Panel	Description
1	CLIENTS	Panel consists of Report View List loaded from QMyClients On select, sets %client$ to client number and links to "CustInfo"
2	PRODUCTS	Panel consists of Report View List loaded from Qproducts On select, passes product number to panel "ProdInfo"
3	INVOICES	Panel consists of Report View loaded from QMyInvoices and Only_Unpaid Drop List. On select, passes invoice number to panel "InvInfo"
4	SALES	Panel consists of Smart Chart from QMyInvoices
5	REVENUE	Panel consists of 3D Column chart with chrt_revenue as default program and load logic in Initialize_chart_1

New PxPlus 2019 feature

DireXions 2019

STOP?

- We could stop right now and have a working product
  - Client list for sales rep
  - Product information with pricing and quantity on hand
  - Invoice list for sales rep
  - Chart of top five clients

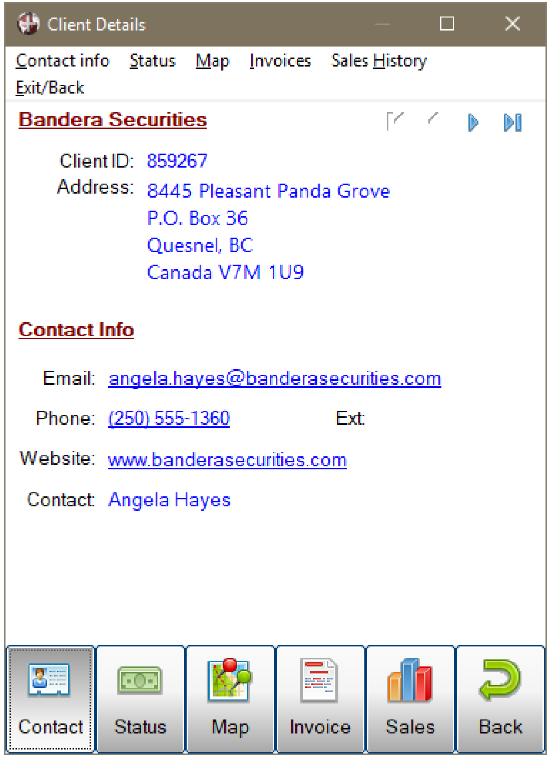
Using only the PxPlus tools
… and very limited coding

CustInfo Panel

- Panel with buttons to select display
  - Contact information
  - Accounting status
  - Location map (via Google maps)
  - Invoices paid and outstanding
  - Sales history

… and a Back button

Client Information Display



Bandera Securities

Client ID: 859267
Address: 8445 Pleasant Panda Grove
P.O. Box 36
Quesnel, BC
Canada V7M 1U9

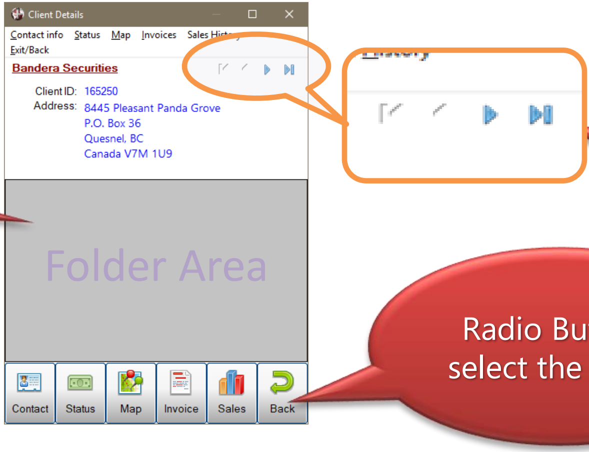
Contact Info

Email: angela.hayes@banderasecurities.com
Phone: (250) 555-1360 Ext:
Website: www.banderasecurities.com
Contact: Angela Hayes

DireXions 2019

CustInfo Panel

Client Folders



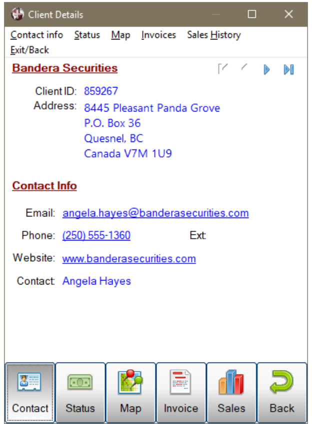
Bandera Securities

Client ID: 165250
Address: 8445 Pleasant Panda Grove
P.O. Box 36
Quesnel, BC
Canada V7M 1U9

"Tabless" Folder used to display information

File Navigation buttons

Radio Buttons select the folder

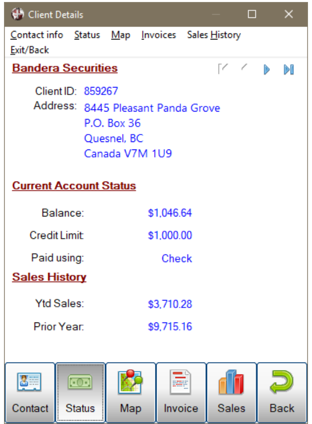


Bandera Securities

Client ID: 859267
Address: 8445 Pleasant Panda Grove
P.O. Box 36
Quesnel, BC
Canada V7M 1U9

Contact Info

Email: angela.hayes@banderasecurities.com
Phone: (250) 555-1360 Ext
Website: www.banderasecurities.com
Contact: Angela Hayes



Bandera Securities

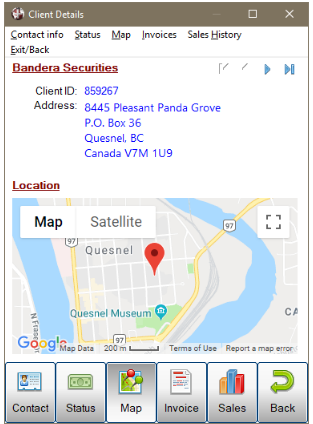
Client ID: 859267
Address: 8445 Pleasant Panda Grove
P.O. Box 36
Quesnel, BC
Canada V7M 1U9

Current Account Status

Balance: $1,046.64
Credit Limit: $1,000.00
Paid using: Check

Sales History

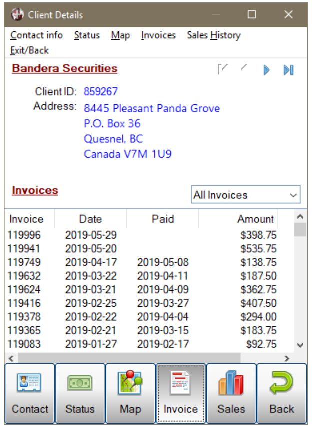
Ytd Sales: $3,710.28
Prior Year: $9,715.16



Bandera Securities

Client ID: 859267
Address: 8445 Pleasant Panda Grove
P.O. Box 36
Quesnel, BC
Canada V7M 1U9

Location

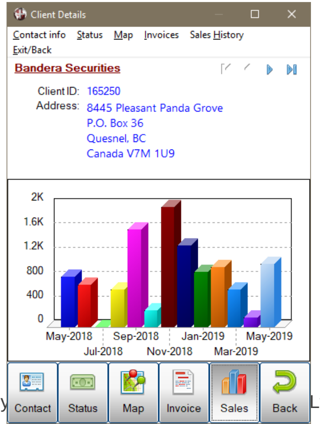


Bandera Securities

Client ID: 859267
Address: 8445 Pleasant Panda Grove
P.O. Box 36
Quesnel, BC
Canada V7M 1U9

Invoices All Invoices

Invoice	Date	Paid	Amount
119996	2019-05-29		$398.75
119941	2019-05-20		$535.75
119749	2019-04-17	2019-05-08	$138.75
119632	2019-03-22	2019-04-11	$187.50
119624	2019-03-21	2019-04-09	$362.75
119416	2019-02-25	2019-03-27	$407.50
119378	2019-02-22	2019-04-04	$294.00
119365	2019-02-21	2019-03-15	$183.75
119083	2019-01-27	2019-02-17	$92.75



Bandera Securities

Client ID: 165250
Address: 8445 Pleasant Panda Grove
P.O. Box 36
Quesnel, BC
Canada V7M 1U9

CustInfo Panel

Folder Assignments

Special *i*Nomads class of buttons

- "Tabless" folder tabs/panels

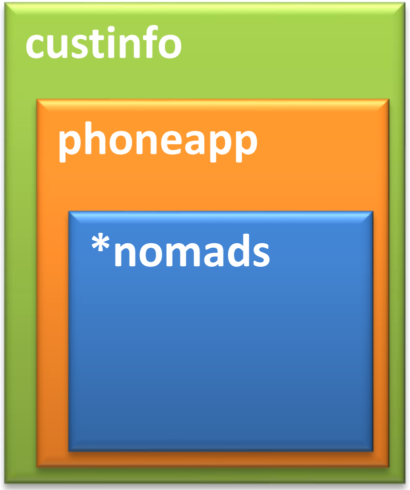
Tab	Panel	Description
1	CONTACT	Panel with buttons whose text will contain email address, phone number, web site and a read only contact name field.
2	CUSTSTAT	Panel consists of read only input fields for balance, credit limit, etc.
3	CUSTMAP	Panel consists of external OCX to Google maps. Sets map type to "roadmap" and zoom factor to 14. *Our API key comes from a ENV variable.*
4	CUSTINV	Panel with drop down for Paid/Unpaid & Report View Smart List loaded from the query "QCustInv“. Triggers are Paid/Unpaid drop down and client.
5	CUSTSALES	Panel consists of 3D Column chart with chrt_cst_sales as default program. Dependency table triggers load when client changes.

Same Folder select logic as MAIN panel

CustInfo Panel Logic

Custinfo Object

- Custom program logic for the Custinfo panel in custinfo.pvc object
  - Bulk of the logic in helper object "phoneapp"
    - custinfo object defines table, key and security logic
    - phoneapp object does most of processing
    - *nomads object interfaces with Nomads/*i*Nomads
  - Custinfo object 'LoadVariables function loads panel data
    - Generate address from individual fields
    - Convert paid by code to plain text from single character
    - Load button text values if required



PhoneApp Object

Common Logic for Displays

- Object contains common code that will be used
  - e.g. what will be displayed by client, product and invoice
  - Inherited by other objects (LIKE directive)
    - Opens table when created
    - Opens Data Dictionary, *if required*
  - Relies on *parent* object definitions

PhoneApp Object

Parent Object Definition

Property/Method	Description
_TableName$	Name of table to be accessed
_FileName$	Pathname of file to be accessed (used if _TableNames$ is empty)
_KeyName$	Name of the key to be used for table/file being accessed
_KeyPrefix$	Value to be inserted in front of any record key when accessing data from table/file
_MapAddress$	Value to be passed as address to Google Maps when a map is present
CanAccess()	Parent should return non-zero value if current record is allowed to be accessed (Zero to skip)
LoadVariables()	Called after record is read and used by Parent to load values into panel fields as required

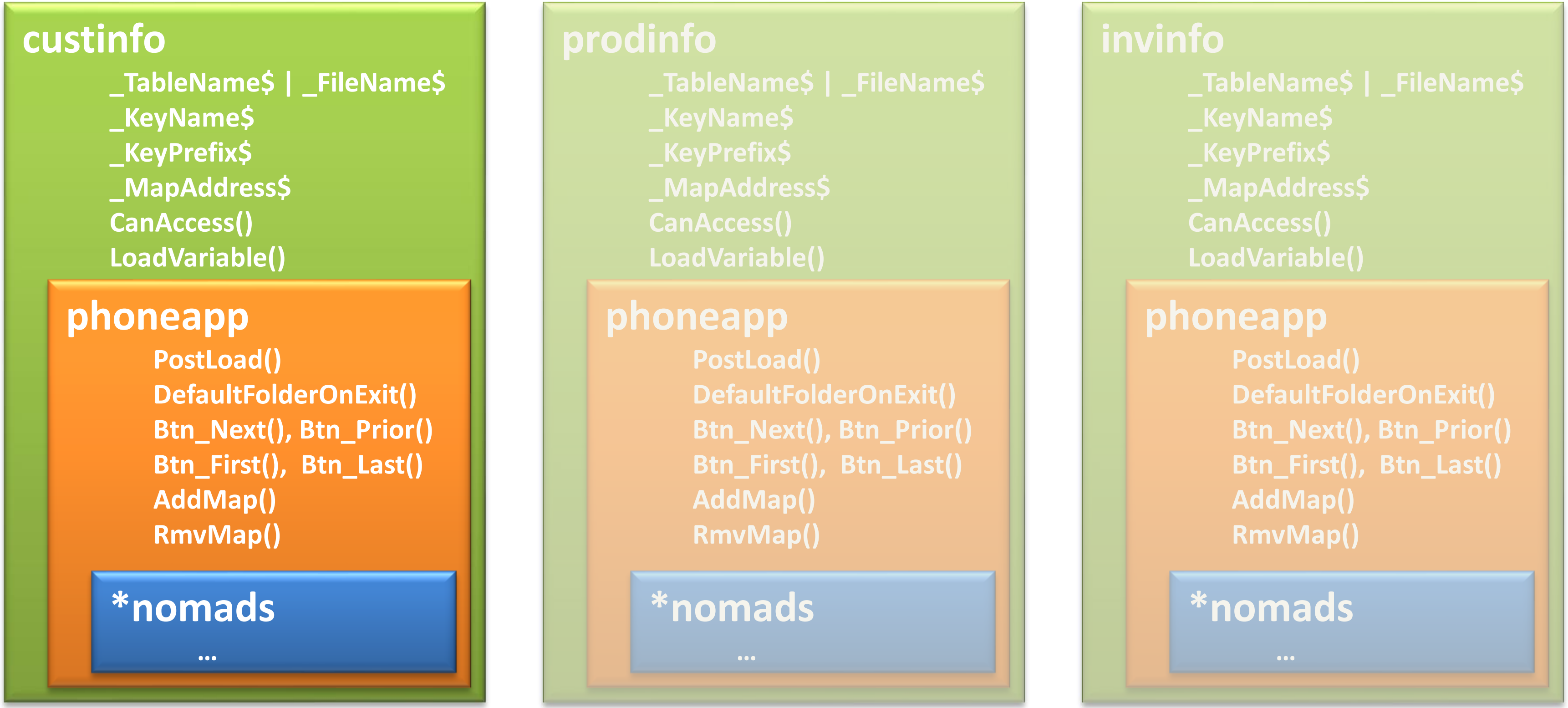
PhoneApp Object

Methods Provided by Object

Method	Description
PostLoad()	Called by Nomads after panel loaded. Method finds first/last records and positions to desired record if passed a starting point.
Btn_Next()	Reads next record. Calls 'CanAccess method to see if allowed to display. If not, skips and reads again.
Btn_Prior()	Reads prior record. Calls 'CanAccess method to see if allowed to display. If not, skips and reads again.
Btn_First()	Reads first record. (Reads null key then reads next)
Btn_Last()	Reads last record. (Reads high value key then reads prior)
AddMap(ctl)	Called to indicate that a map exists on panel and may need updating.
RmvMap()	Called to indicate that a map has been removed.
DefaultFolderOnExit()	Called by Nomads when exiting folder. Removes map references.

These could be better. Should use saved First/Last key

Object Interactions



custinfo

_TableName$ | _FileName$
_KeyName$
_KeyPrefix$
_MapAddress$
CanAccess()
LoadVariable()

phoneapp

PostLoad()
DefaultFolderOnExit()
Btn_Next(), Btn_Prior()
Btn_First(), Btn_Last()
AddMap()
RmvMap()

*nomads

…

prodinfo

_TableName$ | _FileName$
_KeyName$
_KeyPrefix$
_MapAddress$
CanAccess()
LoadVariable()

phoneapp

PostLoad()
DefaultFolderOnExit()
Btn_Next(), Btn_Prior()
Btn_First(), Btn_Last()
AddMap()
RmvMap()

*nomads

…

invinfo

_TableName$ | _FileName$
_KeyName$
_KeyPrefix$
_MapAddress$
CanAccess()
LoadVariable()

phoneapp

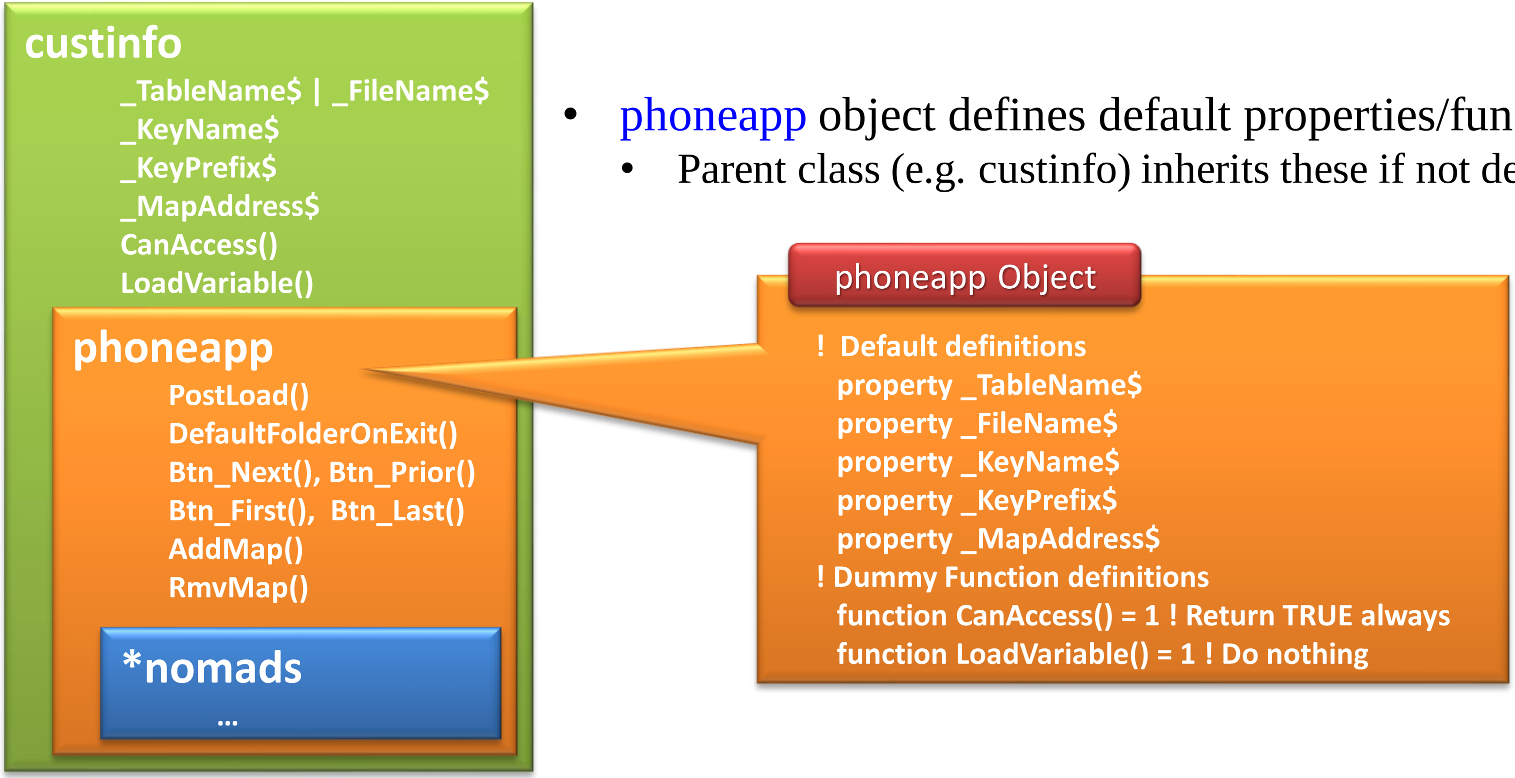
PostLoad()
DefaultFolderOnExit()
Btn_Next(), Btn_Prior()
Btn_First(), Btn_Last()
AddMap()
RmvMap()

*nomads

…

Object Inheritance

- phoneapp object defines default properties/functions
  - Parent class (e.g. custinfo) inherits these if not defined



custinfo

_TableName$ | _FileName$
_KeyName$
_KeyPrefix$
_MapAddress$
CanAccess()
LoadVariable()

phoneapp

PostLoad()
DefaultFolderOnExit()
Btn_Next(), Btn_Prior()
Btn_First(), Btn_Last()
AddMap()
RmvMap()

*nomads

…

phoneapp Object

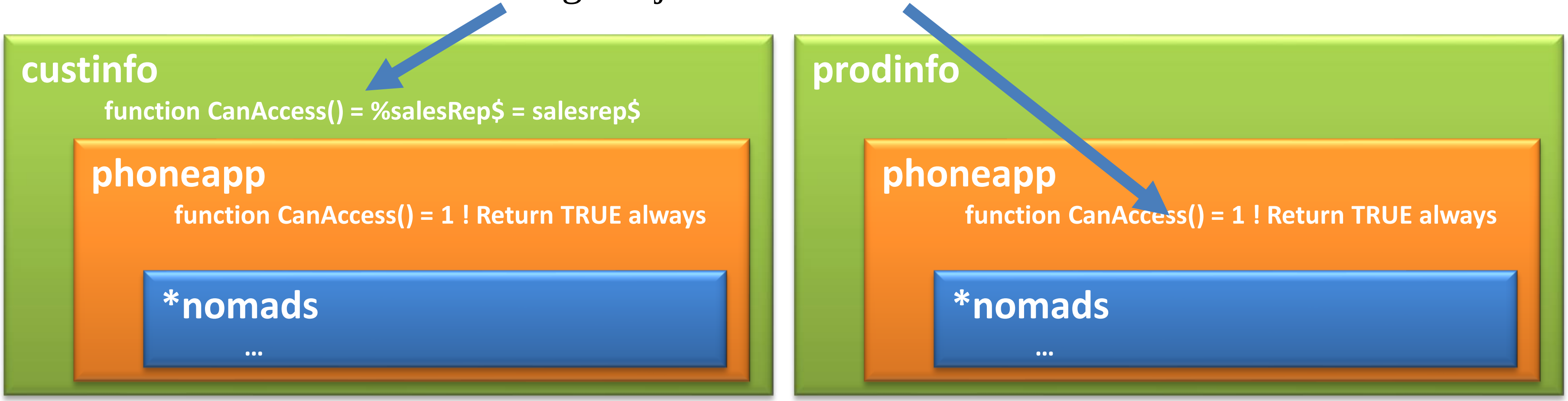
```
!  Default definitions
   property _TableName$
   property _FileName$
   property _KeyName$
   property _KeyPrefix$
   property _MapAddress$
! Dummy Function definitions
   function CanAccess() = 1 ! Return TRUE always
   function LoadVariable() = 1 ! Do nothing
```

Object Inheritance

- Function CanAccess controls if user can view record

```
If NOT( _obj'CanAccess()  ) Then Skip_Record
```

- Method 'cascades' through object inheritance



custinfo

function CanAccess() = %salesRep$ = salesrep$

phoneapp

function CanAccess() = 1 ! Return TRUE always

*nomads

…

prodinfo

phoneapp

function CanAccess() = 1 ! Return TRUE always

*nomads

…

CustInfo Object Review

Panel Object

```
!
! custinfo.pvc – Customer information object
!
 def class "custinfo"
 like "phoneapp"
!
  property _tableName$ = "ClientMaster"
  property _keyName$ = "ByRepName"   ! Clients by Sales rep
  property _keyPrefix$ = pad(%salesRep$,3,$00$) ! Pad to proper key length
  property _MapAddress$ = address1$+", "+city$+", "+state$+", "+country$
!
  function CanAccess() = %salesRep$ = salesrep$ ! Check to see if same sales rep
  function LoadVariables() Do_LoadVariables ! Called after record read to load address and paid by
!
 local address$, payby$
 end def
```

….. Continued next slide

CustInfo Object Review

Panel Object

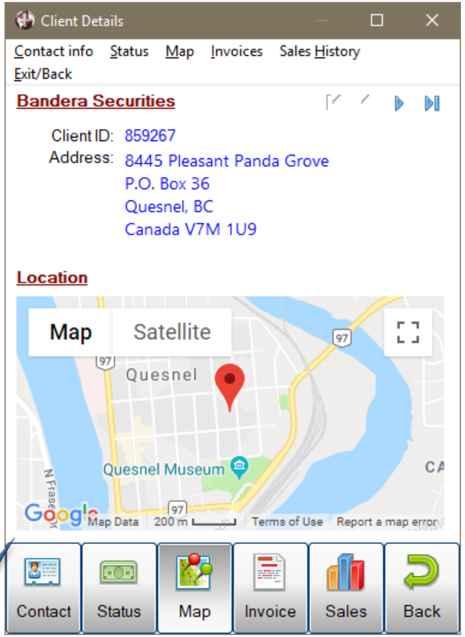
```
Do_LoadVariables:
  %clientid$=clientid$
!
  address$ = address1$+$0A$+address2$+$0A$+city$+", "+state$+$0A$+country$+" "+zipcode$
  address$ = sub(address$,$0A0A$,$0A$)
  if stp(address$,3,*" ,"+$0A$) = "" then address$ = ""
!
  payby$ = tbl(pos(paymenttype$="CVMA"),"Code "+paymenttype$,"Check","Visa","MasterCard","Amex")
!
  if not(has_email)  then return   ! See if email and contact info present
!
  if email.ctl  then email.ctl'text$ = email$ ! Set the text
  if phonenumber.ctl then phonenumber.ctl'text$ = phonenumber$
  if website.ctl  then website.ctl'text$ = website$
  return
```

CustMap Panel

- Displays Google map for an address
  - _MapAddress$ variable set in CustInfo object
    - Must be in the proper address format for Google
  - Panel consists of OCX/COM control
    - PxPlus extension [px]google.maps
    - Post Create Executes:

```
_obj'AddMap(id)
```

- PhoneApp Object handles the rest
  - Change of record causes map to refresh to new address



Bandera Securities

Client ID: 859267

Address: 8445 Pleasant Panda Grove
P.O. Box 36
Quesnel, BC
Canada V7M 1U9

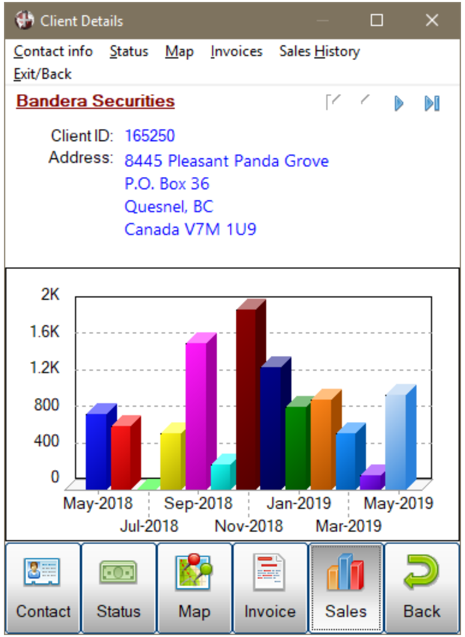
Location

Show Me

CustSales Panel

- Displays monthly sales for the past year
  - Uses program chrt_cst_sales to generate data
    - Program assures data for EVERY month
      - Option under consideration for Smart Charts 2020
  - Panel resets %cur_client$ when loaded
    - Dependency table updates chart when client changes

Sales Chart



Bandera Securities

Client ID: 165250

Address: 8445 Pleasant Panda Grove
P.O. Box 36
Quesnel, BC
Canada V7M 1U9

DireXions 2019

CustSales Panel

Chrt_cst_sales Program

```
!
! chrt_cst_sales -- Update chart with client sales for past 12 months
!
INITIALIZE_CHART_1:
 max_date$ = ""
!
 dim revenues
!
 %cur_client$ = %clientid$
!
 select * from "invhdr"
!
 if %clientid$ <> clientid$ then continue
!
 year$ = mid(invoicedate$,1,4)
 month$ = mid(invoicedate$,5,2)
!
 max_date$ = max(max_date$,invoicedate$)
 revenues[year$+month$] += amount + tax
!
 next record
```

Could use Alt key on Client

```
!
 if max_date$=""  then max_date$=dte(0:"YYYYMMDD")
!
 data$ = "";  labels$ = ""
!
 year = num(max_date$(1,4))-1
 month = num(max_date$(5,2))
!
 for I = 1 to 13
 labels$ += dte(jul(year,month,1):"%Ms-%Y") + ","
 month$ = str(month:"00")
 data$ += "," + str(revenues[str(year:"0000") + str(month:"00")])
!
 month++
 if month>12 then year++;  month = 1
 next
!
 chart load chart_1.ctl,data$(2)+"/"
 chart_1.ctl'pointtext$=labels$
 chart_1.ctl'colorByPoint=1
!
 return
```

Chart separator set to comma

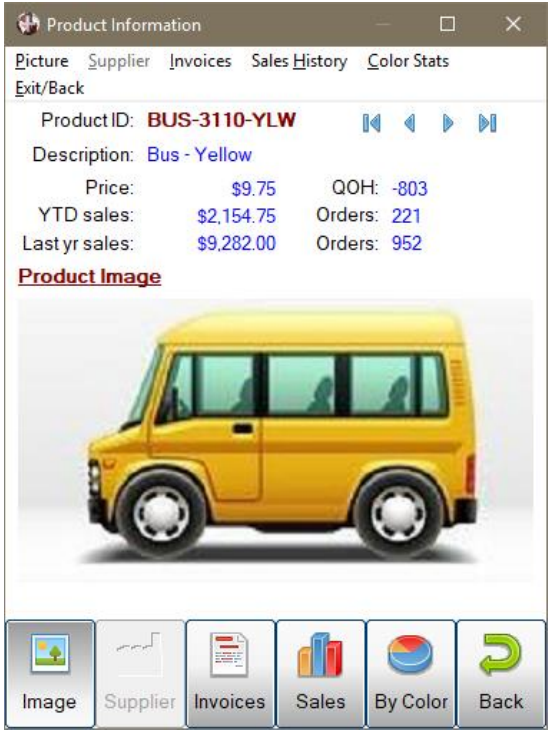
Program assures value exists EVERY month

ProdInfo Panel

- Panel with buttons to select display
  - Product image
  - Supplier information
  - Invoices for product
  - Sales history for product
  - Sales by color analysis

… and a Back button

Product Information Display



Product ID: BUS-3110-YLW

Description: Bus - Yellow

Price: $9.75 QOH: -803

YTD sales: $2,154.75 Orders: 221

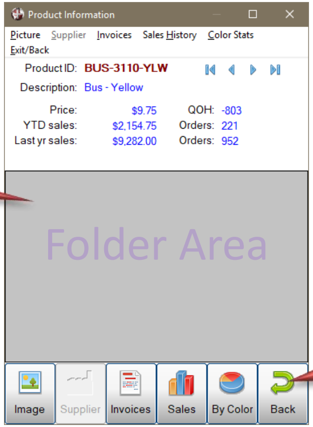
Last yr sales: $9,282.00 Orders: 952

Product Image

ProdInfo Panel

Product Folders

"Tabless" Folder used to display information



Product ID: BUS-3110-YLW

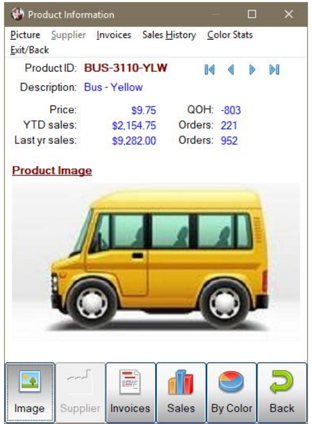
Description: Bus - Yellow

Price: $9.75 QOH: -803

YTD sales: $2,154.75 Orders: 221

Last yr sales: $9,282.00 Orders: 952

Radio Buttons select the folder



Product ID: BUS-3110-YLW

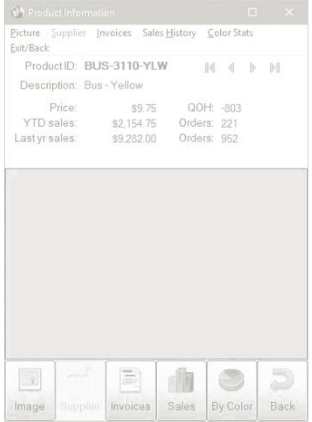
Description: Bus - Yellow

Price: $9.75 QOH: -803

YTD sales: $2,154.75 Orders: 221

Last yr sales: $9,282.00 Orders: 952

Product Image



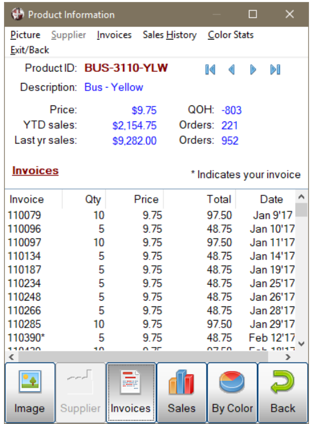
Product ID: BUS-3110-YLW

Description: Bus - Yellow

Price: $9.75 QOH: -803

YTD sales: $2,154.75 Orders: 221

Last yr sales: $9,282.00 Orders: 952



Product ID: BUS-3110-YLW

Description: Bus - Yellow

Price: $9.75 QOH: -803

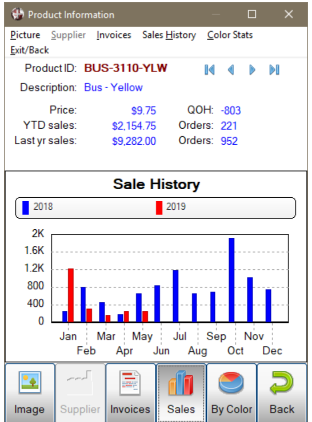
YTD sales: $2,154.75 Orders: 221

Last yr sales: $9,282.00 Orders: 952

Invoices

* Indicates your invoice

Invoice	Qty	Price	Total	Date
110079	10	9.75	97.50	Jan 9'17
110096	5	9.75	48.75	Jan 10'17
110097	10	9.75	97.50	Jan 11'17
110134	5	9.75	48.75	Jan 14'17
110187	5	9.75	48.75	Jan 19'17
110234	5	9.75	48.75	Jan 25'17
110248	5	9.75	48.75	Jan 26'17
110266	5	9.75	48.75	Jan 28'17
110285	10	9.75	97.50	Jan 29'17
110390*	5	9.75	48.75	Feb 12'17



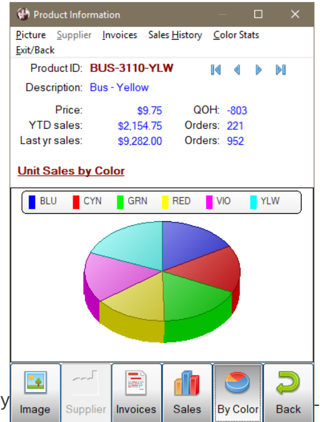
Product ID: BUS-3110-YLW

Description: Bus - Yellow

Price: $9.75 QOH: -803

YTD sales: $2,154.75 Orders: 221

Last yr sales: $9,282.00 Orders: 952



Product ID: BUS-3110-YLW

Description: Bus - Yellow

Price: $9.75 QOH: -803

YTD sales: $2,154.75 Orders: 221

Last yr sales: $9,282.00 Orders: 952

Unit Sales by Color

ProdInfo Panel

Folder Assignments

- "Tabless" folder tabs/panels

Tab	Panel	Description
1	PRODIMAGE	Panel with image of product
2	---	Empty option – could be for supplier information
3	PRODINV	Panel with Report View List of invoices for product. Loaded using Smart Load of query "QProdInv". Triggered on change of ProductCode$.
4	PRODSALES	Panel with drop down for Paid/Unpaid and report view Smart List loaded from the query "QCustInv". Triggers are Paid/Unpaid drop down and client.
5	PRODCOLOR	Panel consists of 3D Column chart with chrt_cst_sales as default program. Dependency table triggers load when client changes.

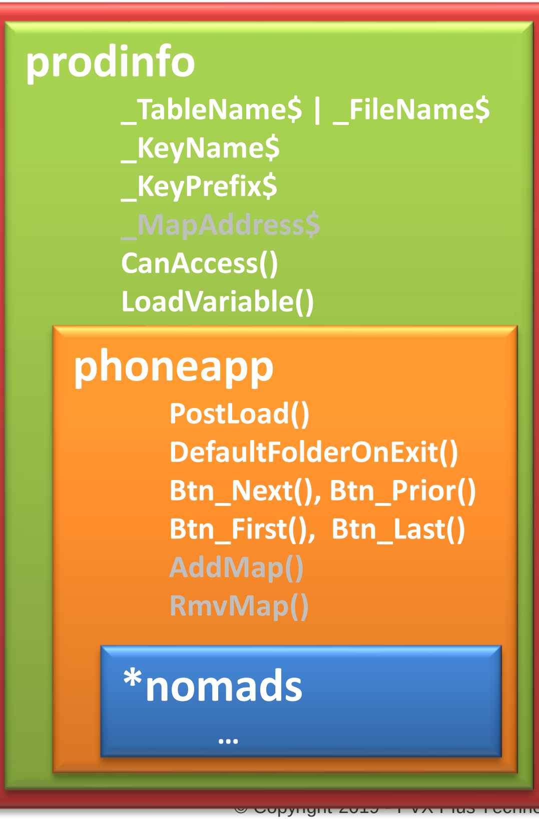
Same Folder select logic as MAIN panel

ProdInfo Object

```
!
! prodinfo.pvc – Product information object
!
  def class "prodinfo"
  like "phoneapp"
!
  property _tableName$ = "ProductMaster"
  property _keyName$ = ""
  property _keyPrefix$ = ""
!
  function LoadVariables() Do_LoadVariables
!
  local imagepath$
  end def
!
Do_LoadVariables:
  %productcode$ = productcode$
  imagepath$ = "images/products/"+lcs(productcode$)+".jpg"
!
  return
```

Data field names matched panel field names

Panel Object



prodinfo
_TableName$ | _FileName$
_KeyName$
_KeyPrefix$
_MapAddress$
CanAccess()
LoadVariable()

phoneapp
PostLoad()
DefaultFolderOnExit()
Btn_Next(), Btn_Prior()
Btn_First(), Btn_Last()
AddMap()
RmvMap()

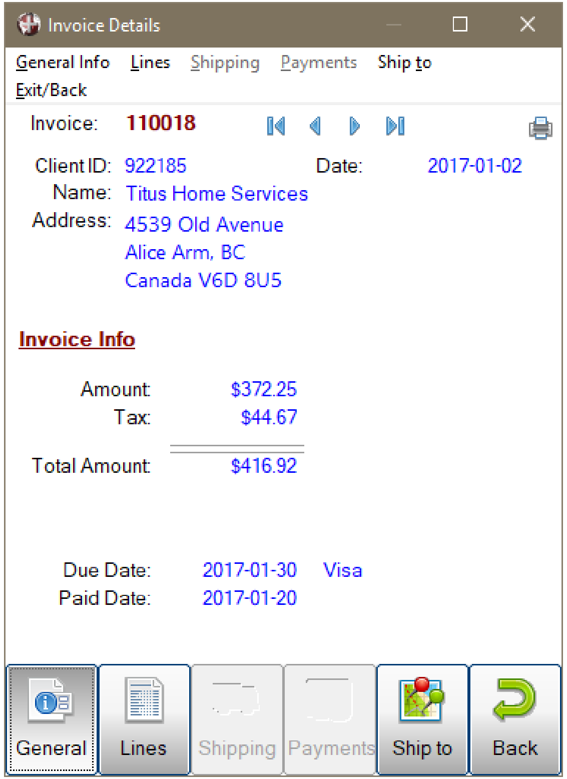
*nomads
…

InvInfo Panel

- Panel with buttons to select display
  - General information
  - Lines on the invoice
  - Shipping information
  - Payment information
  - Shipping map

… and a Back button

Invoice Information Display



Invoice: 110018

Client ID: 922185 Date: 2017-01-02
Name: Titus Home Services
Address: 4539 Old Avenue
Alice Arm, BC
Canada V6D 8U5

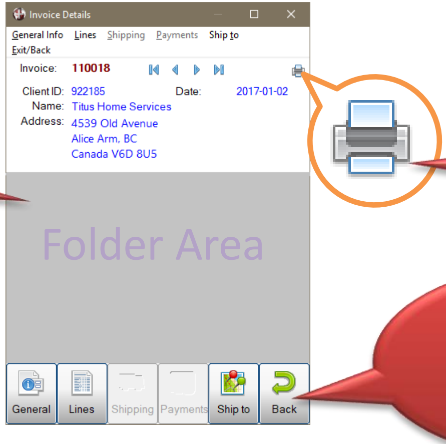
Invoice Info

Amount: $372.25
Tax: $44.67
Total Amount: $416.92

Due Date: 2017-01-30 Visa
Paid Date: 2017-01-20

InvInfo Panel

Invoice Folders



Invoice: 110018

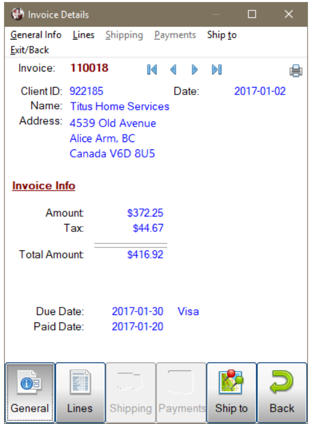
Client ID: 922185 Date: 2017-01-02
Name: Titus Home Services
Address: 4539 Old Avenue
Alice Arm, BC
Canada V6D 8U5

Folder Area

"Tabless" Folder used to display information

Print Invoice Button

Radio Buttons select the folder



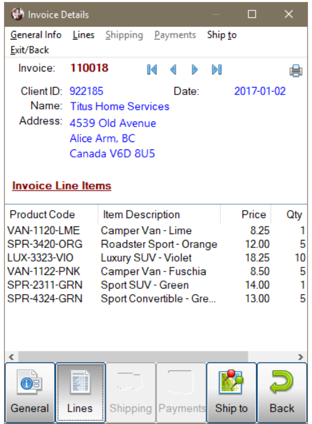
Invoice: 110018

Client ID: 922185 Date: 2017-01-02
Name: Titus Home Services
Address: 4539 Old Avenue
Alice Arm, BC
Canada V6D 8U5

Invoice Info

Amount: $372.25
Tax: $44.67
Total Amount: $416.92

Due Date: 2017-01-30 Visa
Paid Date: 2017-01-20

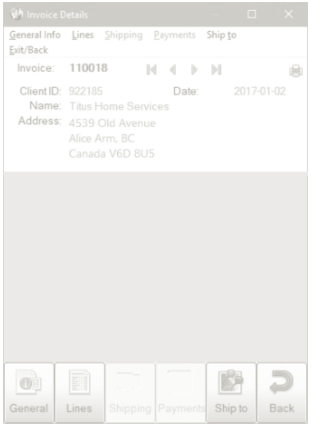


Invoice: 110018

Client ID: 922185 Date: 2017-01-02
Name: Titus Home Services
Address: 4539 Old Avenue
Alice Arm, BC
Canada V6D 8U5

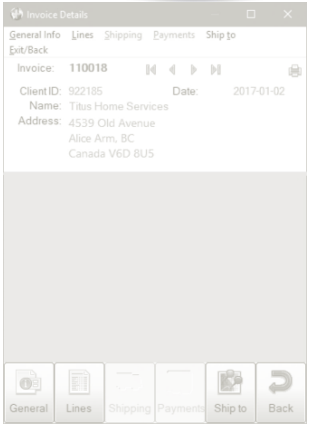
Invoice Line Items

Product Code	Item Description	Price	Qty
VAN-1120-LME	Camper Van - Lime	8.25	1
SPR-3420-ORG	Roadster Sport - Orange	12.00	5
LUX-3323-VIO	Luxury SUV - Violet	18.25	10
VAN-1122-PNK	Camper Van - Fuschia	8.50	5
SPR-2311-GRN	Sport SUV - Green	14.00	1
SPR-4324-GRN	Sport Convertible - Gre...	13.00	5



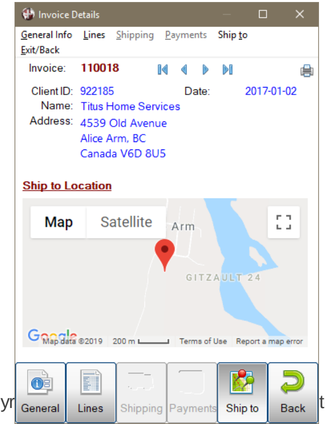
Invoice: 110018

Client ID: 922185 Date: 2017-01-02
Name: Titus Home Services
Address: 4539 Old Avenue
Alice Arm, BC
Canada V6D 8U5



Invoice: 110018

Client ID: 922185 Date: 2017-01-02
Name: Titus Home Services
Address: 4539 Old Avenue
Alice Arm, BC
Canada V6D 8U5


Invoice Details
General Info Lines Shipping Payments Ship to
Exit/Back
Map
Satellite
Arm
GITZAULT 24
Google
Map data ©2019 200 m Terms of Use Report a map error
General
Lines
Shipping
Payments
Ship to
Back

Invoice: 110018

Client ID: 922185 Date: 2017-01-02
Name: Titus Home Services
Address: 4539 Old Avenue
Alice Arm, BC
Canada V6D 8U5

Ship to Location

InvInfo Panel

Folder Assignments

- "Tabless" folder tabs/panels

Tab	Panel	Description
1	INVHEADER	Panel with read only data fields which contain invoice totals and dates.
2	INVLINES	Panel consists of report view loaded using Smart load of QInvLines query.
3	--	Empty option – could be for shipping information.
4	--	Empty option – could be for payment information.
5	INVSHIPTO	Panel consists of external OCX to Google maps. Sets map type to "roadmap" and zoom factor to 14.

Same Folder select logic as MAIN panel

InvInfo Object

Panel Object

```
! invnfo.pvc -- Invoice information object
!
! If Arg_2$ is set when panel invoked, it must contain client id to filter.
! If %salesrep$ is set then access by sales rep.
!
  def class "invinfo" create required delete required
  like "phoneapp"
!
   property _tableName$="InvoiceHeaders"
   property _MapAddress$=address1$+", "+city$+", "+state$+", "+country$
!
   function CanAccess() = (%salesRep$=salesrep$) and ((arg_2$="") or (arg_2$=clientid$))
   function LoadVariables()Do_LoadVariables ! Called after record read to load address and payby
!
   function Btn_Print() Do_Print ! Print the invoice
!
   local address$,payby$
   local total_amt, client_fileno, clientname$
!
 end def
```

Must be current sales rep and if client specified (arg_2$) it must match

Print invoice button

InvInfo Object

Panel Object (continued)

```
on_create:
  open (hfn,iol=*) table "Clientmaster"
  client_fileno=lfo
!
  exit
!
 on_delete:
  close (client_fileno)
  exit
!

 Do_LoadVariables:
!
  cst.clientname$="<Unknown client>"
  read (client_fileno,key=clientid$,dom=*next,rec=cst$)
  clientname$=cst.clientname$
!
  %clientid$=clientid$
```

```
!
 address$ = address1$ + $0A$ + address2$ + $0A$ +  city$
    + ", " + state$ + $0A$ + country$ + " "+zipcode$
  address$ = sub(address$,$0A0A$,$0A$)
!
  if stp(address$,3,*" ,"+$0A$) = "" \
    then address$ = ""
!
  total_amt = amount+tax
  payby$ = tbl(pos(paymenttype$="CVMA"), "Code" +
    paymenttype$,"Check","Visa","MasterCard","Amex")
!
  return
```

InvInfo Object

Panel Object (continued)

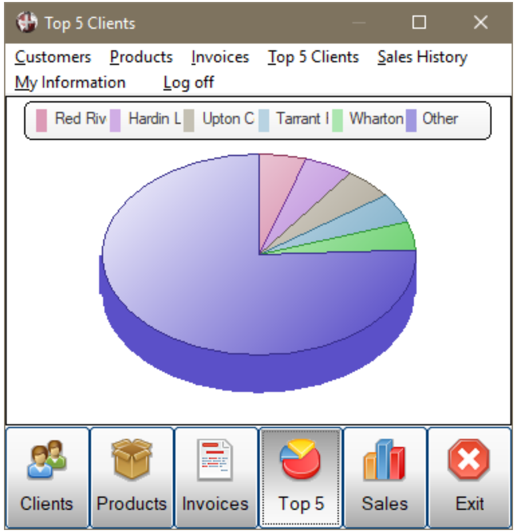
```
! --------
! Do_Print: Print the invoice in response to Print button
!
Do_Print:
  RPT = new("*rpt/pvxreport" for program)
!
  If RPT'open("reports/invoice.pvr") = 0 \
   then msgbox "Unable to locate invoice layout","System error","!";
          return
!
  open (hfn)"*viewer*"
  PRT=lfo
!
  RPT'outputprint(PRT)
  RPT'SetParameter("InvNum",InvoiceNumber$) ! Set the InvNum parameter
  RPT'runreport()
  RPT'close()
  close (PRT)
!
  return
```

Report Writer used
to generate
Invoice

Sales Panel (Top 5 Clients)

Main Panel

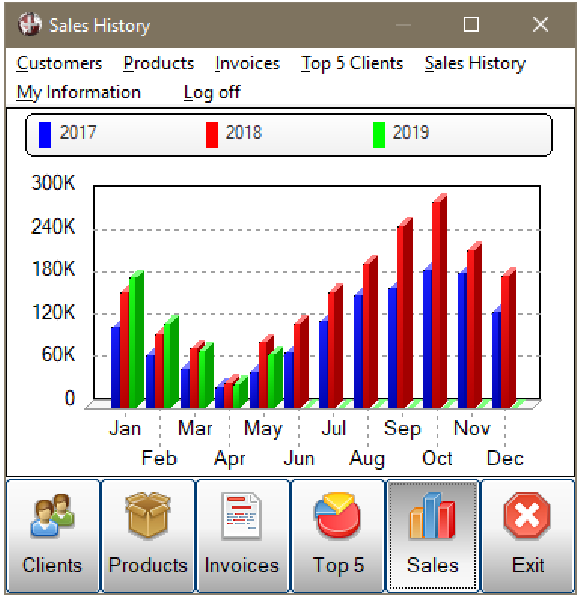
- Panel contains chart
  - 3D pie chart using 'Plus' engine
  - Smart Charts used to load data
    - Query source "QMyInvoices"



Revenue Panel

Main Panel

- Panel contains chart
  - 3D Column chart using 'Plus' engine
  - Data loaded by "chrt_revenu" program
    - Similar to chrt_cst_sales
    - Made sure all months for last two years exist



DireXions 2019

Running on the Web

- Created *i*Nomads transaction "demo-rep"
  - Starts in *demos/salesrep directory
  - Run panel "logon" in "scrnlib.en"
  - Uses template "phonedark" (or "phoneapp" for white background)
    - Custom templates designed for use with smart phones
      - Dark grey title background with bold white letters; font size 20
      - Panel font height 18, width 10

Running on the Web

- PhoneApp and PhoneDark template settings
  - Overlay windows as opposed to Popup plus force resize on load
  - Two custom META tags in HTML settings for Template options
    - Define the desired width of the smartphone (426 ~ 42 * 10 pixels per character wide)

```
<meta name="viewport" content="width=426">
```

      - Phone will automatically scale panel to this width
      - Length will vary based on smartphone

    - Set smartphone to ignore numeric text that looks like phone number

```
<meta name="format-detection" content="telephone=no">
```

      - Primarily for iPhones which may confuse invoice numbers for phone numbers

Custom Style Sheet (style.css)

Tweaks to the UI to improve look

```
/* Style sheet for phonedark */
.iCopyright, .iTtlBtn { display: none; }     /* Hide copyright and title buttons – leaves iNomads release info */
#Sect_Pnl {max-width: 426px; margin: 0 auto; }  /* Forces maximum panel width */

/* Dark color settings */
body { background-image: radial-gradient(#555, #000); color: White; }
.iLv { background-color: transparent; }
.iMbxDiv, .iLvHRow, .iMenuDiv { color: Black; }
div.iChrt { color: silver; }

/* Any Dark red text become bright red, Dark Blue becomes Cyan */
*[style*="color:#800000"] { color: Red !important; }
*[style*="color:#0000ff"] {color: Cyan !important; }

/* Tweaks */
body.on_ipad  button, body. on_iphone button { margin: 0; padding: 2px; }   /* Safari  rounds buttons with wide margins
div[data-ctl="btn_bar"] .btTxt { font-size: 85%; }     /* Smaller text on Button bar */

/* Spot light effect for photos (set in iNomads class for image) */
div.spotlight { border-radius: 50%; border: solid 1px #555; background: radial-gradient(White, Silver); overflow: hidden; }
```

These settings specific for Dark background

What About My Application?

- Our main goal was to provide functionality that is customizable

  - Change the file and field names
    - Adjust keys used for lookup
    - Create work files if need be for non-existent key fields
      - i.e. Sales rep → Client or Client → Invoice

  - Obtain your own Google Maps API key if desired
    - Set in PXP_MAP_KEY environment variable or hard code in panels

  - Change data on the screen depending on need and relevance

Proof of Concept

Sample Infor FACTS

- Demo on our website
  - Using Facts 9.2 demo data set
  - Code/Panel changes included in *demo/2019/phone_facts *(no data files included)*
    - "setup" program needs to point to Facts directory *(should contain providex.ddf)*
      - Uses environment variable Facts_dir for pathname

- Basic changes:
  - Logon screen - Presented company then sales rep
  - *xxxx*Info objects - Changed file/table names and keys
  - Panels - Changed some fields
  - Invoice lists - Changed to offer two choices – Orders / Invoices
  - Products - Added product images

Logic derived from what we could 'guess' from data dictionary

Proof of Concept

Sample Sage 100

- Demo on our website
  - Using Sage 100 demo data set
  - Code/Panel changes included in *demo/2019/phone_sage *(no data files included)*
    - "setup" program needs to point to Sage 100 "MAS90" directory *(should contain providex.ddf)*
      - Uses environment variable Sage100_dir for pathname

- Basic changes:
  - Logon screen - Presented company, division then sales rep
  - Work files - Created work files to speed up some key search (i.e. customers could have multiple sales reps)
  - *xxxx*Info objects - Changed file/table names and keys
  - Panels - Changed some fields

Logic derived from what we could 'guess' from data dictionary

Goal of this Presentation

To provide guidance on the development of PxPlus Web based applications targeting Smartphones.

Hopefully, we succeeded.

Additional Resources

The help link(s) below refer to the current on-line help pages. The functionality may have been further updated since the PxPlus 2019 (version 16) release.

- C.A.T. App Demo
(best viewed on a smartphone)
- iNomads Transactions
- iNomads Templates
- Dependency Definitions
- Smart Controls
- Concurrent Panels
- Object Oriented PxPlus
- Google Maps Interface
- Report Writer