Document Title:	RACKETS: Case Tracking and Mapping System
Author(s):	Colin Reilly ; Victor Goldsmith
Document No.:	182918
Date Received:	June 14, 2000
Award Number:	98-LB-VX-0004

Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

RACKETS

Case Tracking and Mapping System

Developed for:
United States Attorney's Office,
Southern District of New York

Developed by:
Colin Reilly
Center for Applied Studies of the Environment,
Hunter College, City University of New York
Victor Goldsmith,
Dean of Research,
Hunter College, City University of New York

FINAL REPORT TO:
National Institute of Justice
United States Department of Justice
Grant No. 98-LB-VX-0004
August 1999

CENTER FOR APPLIED STUDIES OF THE ENVIRONMENT CITY UNIVERSITY OF NEW YORK

Professor Victor Goldsmith (Hunter College), Director
Professor Arthur Langer (Brooklyn College), Deputy Director
Professor Robert Graff (City College), Deputy Director

Senior Colleges of CUNY:
BC - Brooklyn College, CC - City College, GSUC - Graduate School and University Center,
HC - Hunter College, LC - Herbert H. Lehman College and QC - Queens College.

N
QC
BC
0 5
kilometers
1
3
4
5
7
8
9
10
11
12
13

A SPOT Image (June 25, 1988) is shown merged with an EOSAT IMAGE (August 30, 1990), resulting in a multispectral image with a 10 m resolution (after a technique by Ahearn, 1993).In this classfication, the lower numbers represent more urbanized areas. This merged image is from a report prepared for the NYC Department of Environmental Protection (Goldsmith et al., 1993).

Introduction

Rackets is a case tracking and mapping application developed by the Center for Applied Studies of the Environment (CAPSE) for the U.S. Attorney's Office (USAO), Southern District of New York (SDNY), hereto referred to as SDNY. The SDNY and the National Institute of Justice (NIJ) provided the funding for this project. Work on the project began in July 1997 and the bulk of which ended in October of 1998.

The impetus for Rackets developed out of a desire by the SDNY to move away from the traditionally episodic way of handling cases to a comprehensive and strategic method. The SDNY is the federal prosecuting office for Manhattan and Bronx counties in New York City, as well as Westchester and several other northern counties. As such, the Office's primary mission is to investigate and prosecute federal crimes in these counties. In the course of this work, the Office interacts with many different federal, state and local agencies. Oftentimes, information gathered by one agency for a certain case is related to other cases under investigation by other agencies without either agency aware of the other's investigation. For example, the Department of Agriculture may be investigating Food Stamp fraud in a grocery store where the Bureau of Alcohol Tobacco and Firearms is investigating a gun trafficking ring and there may be a critical link that could strengthen both cases.

Additionally, criminals are linked by geography and pedigree which cross investigative boundaries. Therefore, with the development of system that could collect case specific information and link it to specific geographic locations the Office would have at its disposal the ability to cut across agency lines and establish geographic connections between criminals and federal cooperators. Rackets offers, at a glance, an in depth analytic capacity of the entire federal landscape for the Southern District of New York.

Rackets was initially designed to collect information on the cases, defendants, investigations, wiretaps and warrants for the violet gangs, narcotics, organized crime and general crimes units. The system was developed to collect information either not handled, or not handled at sufficient enough detail by the Office's existing case management system (PROMIS at the time of development). The current case management system, LIONS, is charged with handling the administrative details of the Office's cases and as such does not attempt to collect the detailed information that exists in the charging instruments or through proffers or cooperation agreements. Information is collected on a case by case basis, filed, and utilized further if an office-wide request is made. Rackets attempts to fill this information void by utilizing the information for future investigations and cases. Further, all geographic information is entered and mapped thereby linking the data to discrete locations. Then, the data can be searched by traditional methods or by spatial queries.

The System

Rackets is an end-user application designed to be run largely by the Office's non-technical staff. To accomplish this, a simple user-interface was developed to allow the staff to intuitively enter and query the data. Numerous interactive canned queries and reports are included in the application. The mapping component of the system has also been customized to allow the staff to perform spatial queries without having to be a Geographic Information Systems (GIS) expert. Nonetheless, the system is not maintenance free and requires a degree of technical expertise to maintain and undertake the more complicated tasks.

Information input into the database is collected off of the Office's charging instruments. These documents are digested for all pertinent information and then entered into the system by USAO staff. To clothe this data, pedigree information for all arrested defendants is obtained via electronic transfer from the U.S. Marshals Prisoner Tracking System (PTS) database and entered into Rackets.

As part of the implementation of the system, existing case data from PROMIS were imported into Rackets. This was a cumbersome procedure that required the data to be cleaned (removal of duplicates and closed cases) and converted into relational format. Additionally, prisoner data from PTS was also imported into Rackets. Thus, upon Rackets inception over 2,00 cases and 4,000 defendants resided in the database.

Due to the sensitive nature of some of the data as well as organizational constraints, the system was developed as a stand-alone application. In the future, a link could be developed to LIONS, but until then, the system functions as its own entity. To further maintain security, the system runs on a secure Windows NT network with restricted access.

The Network

The network contains a server (Bronx) and two workstations (New York, and Richmond). Bronx contains all server applications and the data. New York is the GIS workstation and Richmond is for data entry. Both workstations have a mapped-out network drive (Z:\) which links them to the network drive on Bronx.

Rackets

Rackets is a client-server application composed of database and GIS components. A client-server application separates the data and interface components into two separate applications, a client and a server. The server database manages the data while the client applications manage the user interface (i.e., all forms, queries, reports and underlying code). A single copy of the server database exists and is installed on the network server.[1] Operationally, the client applications send requests to the server and in turn the server returns the data or pointers to that data. The database server is the component that carries out the actual updating of the data. This allows multi-users to view and edit the same data

[1] A network server is a physical computer that functions as the network's brain and allows the sharing of resources between computers, while a database server is an application that manages and stores the data of a client-server database.

concurrently, although not the same record. The database was developed using Microsoft Access.

In actuality, Rackets is composed of three database servers that are all linked to the client applications installed on each computer. In total, there are five separate components making up the entire application, which are as follows:

1. Main database server
2. PTS database server
3. Archive database server
4. Database client applications, installed on each workstation.
5. GIS mapping component

The following section discusses the database structure and provides a detailed description of each component, all the objects contained in them and their interconnectivity.

System Components

Database Structure

Since the office carries out its operations on a case basis, the database mimics this organizational structure and uses a case table as its central table. The case table collects information specific to the case, such as the investigating agency and agent, Assistant U.S. Attorney (AUSA) handling the case and pertinent dates. From this table, defendant, wiretap, warrant and target (individuals or entities being investigated) tables are linked and contain from zero to many entries for each case. The next major table stores defendant information and also has other tables tables linked to it. For instance, alias, statutes, and events tables are linked in a one-to-many relationship with the defendant table. Although there is frequently a defendant linked to more than one case, warranting a many-to-many relationship between the case and defendant table, the Office preferred to maintain a one-to-many relationship. This is because many times defendants give different information each time they are arrested. Therefore, all details from each arrest would be recorded arrest those details would be picked up.

Although new arrests are electronically transferred on a daily basis to the office from PTS (detailed in Appendix B), these records are not immediately transferred into the main database. This is due in part because each time there is some activity for a prisoner (e.g., sentencing), update information is electronically transferred to the SDNY. As such, the electronic transfers represent not only new arrestees, but existing prisoners who could predate the system. Rackets is setup to enforce referential integrity (e.g., a defendant record can not be entered without an existing case record to which to be associated) and prevent the accumulation of orphan records (i.e., unrelated records).

Additionally, by maintaining these data in a separate location it enables the Office's staff to monitor all new entries into the system. For this reason, the PTS database functions as a sort of holding cell that contains all of the PTS transfers. These records are maintained in this database until they are transferred to the main database. Even then, only some of the data is transferred to the main database, including the identification number for each prisoner record (pr_key), which allows linkages to all the data remaining in the PTS database for each prisoner. Finally, an archive database server was developed which is an abbreviated version of the main database and stores records of closed cases and sentenced defendants.

Database Servers

As the data manager component, the server database acts as the warehouse of the application and as such stores all the data, and contains the necessary tables to efficiently manage these data. Appendix A is a data dictionary that details every table and field included in the database. All tables in the server application are linked to the client applications from the client end. The file locations of the server applications on Bronx are:

Main - z:\network\rackets\data\Rackets_be.mdb
PTS - z:\network\Pts\snpts,mdb

The client applications on the workstations must be directed to the location of the server via their network drives to the following location:

z:\Rackets\data\Rackets_be.mdb

If changes are made to the location of the server application, the links from the client application need to be updated. To update the table links go to the Linked Table Manager from the Tools menu, select Add-Ins. Once there, separately select all tables from **Rackets_be.mdb** and **snpts.mdb**, and direct the application to their respective file locations.

If a table is accidentally deleted or missing, it can be re-linked from the client application as follows: under the File menu go to Get External Data and select Link Tables. Then direct the application to the location of the table (either Rackets_be.mdb or snpts.mdb) that should be re-linked.

Client

As stated previously, the client application contains the interface objects of the database application. This is what the user sees when running RACKETS from the desktop by double clicking the icon in figure 1.1. The client application contains all the forms, queries, reports, macros, table links and code necessary to enter, navigate through and query the data (the Visual Basic Scripts behind the forms are in Appendix D). A copy of the client application is installed locally on each computer on the network. The local client applications access the server application on the network server via their network drive (Z:\).

Figure 1.1

Mapping Application

The mapping application was developed using Environmental Systems Research Institute's (ESRI) ArcView 3.0a GIS. ArcView was customized to link to the data maintained in the server databases, to automate the Attorney's Offices most common queries and the ability to run simple analyses. The customization includes the following:

- a new menu item Rackets, (see figure 1.2) containing the customized operations;
- a button that labels streets and RACKETS themes (themes based on tables in the RACKETS database);
- a button that saves a map layout as a graphic image file;
- automatic connection to, and visualization of the data in the databases;
- setup of all basemap themes.

The customization is contained in an ArcView project (rackets.apr) and is installed locally on each computer. This project includes all the Avenue code (the scripting language included with ArcView), dialogs, and formatting required to carry out all operations (Appendix C). The mapping application is accessed from the main menu of Rackets (see figure 1.)

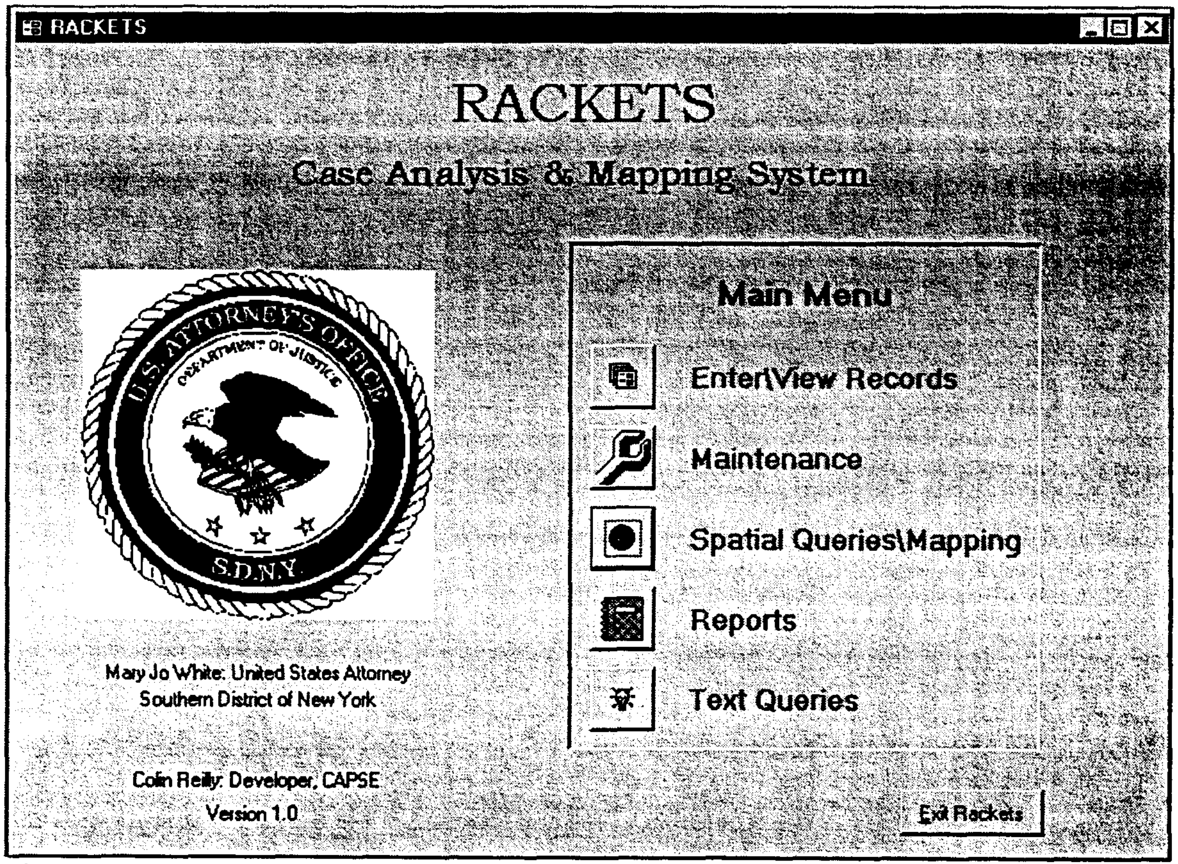

RACKETS
RACKETS
Case Analysis & Mapping System
Main Menu
Enter\View Records
Maintenance
Spatial Queries\Mapping
Reports
Text Queries
Mary Jo White: United States Attorney
Southern District of New York
Colin Reilly: Developer, CAPSE
Version 1.0
Exit Rackets

Figure 1.2

The mapping application is connected to the database via Open Database Connectivity (ODBC). A DSN system data source was set up on every computer. The DSN is referred to by the same name as the application, RACKETS. The connection is made each time the mapping application is run via the "StartUp" script associated with the project, and the connection is subsequently closed each time the project is closed by the "ShutDown" script.

New York functions as the GIS workstation and the GIS functionality has been optimized for this computer, therefore all mapping work should be done on the New York. However the mapping application functions on all computers except for the reporting capability.

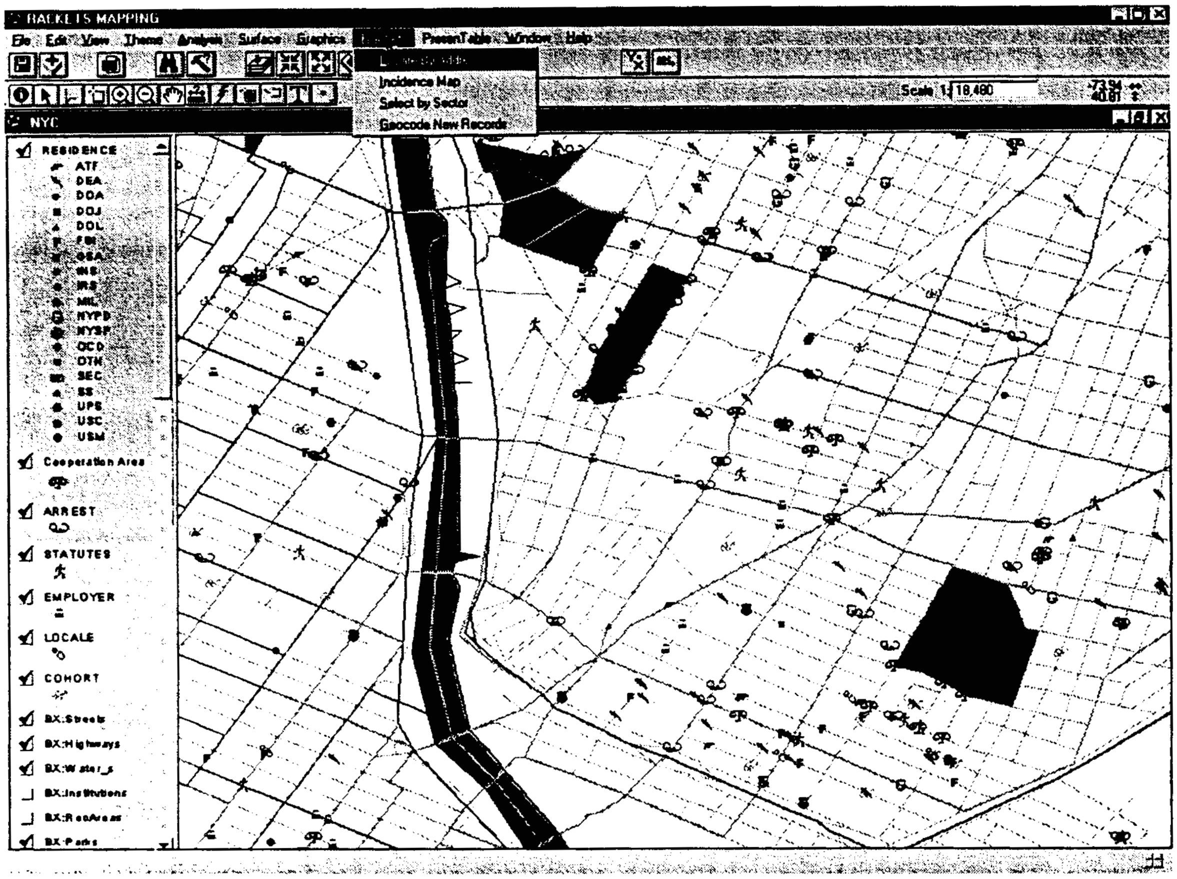


Figure 1.3

RACKETS Themes

These are the themes that are based on tables in the Rackets database. These themes are not ArcView shape files; they are event themes based on virtual tables resident in the Rackets database. These virtual tables are established by sending queries to the database via the ODBC connection and represent a view of the data, not an actual copy of the data. This way, the database manages and stores all the data in one single location and the GIS can map, query and update the data via the ODBC connection. Therefore, one does not need to worry about having to create a shape file from the tabular data each time a new query is run. This prevents the possibility of having numerous time-dependant copies of the data floating around. With the virtual tables, changes made to the data are reflected in the virtual tables immediately after the tables are refreshed. This occurs each time the application is run or by selecting refresh from the table menu for each table.

For each table in the Rackets database that is mapped there are two virtual tables in the mapping application. One is the actual table to be mapped, the map table, and consists of records already geocoded and containing their x and y coordinates (e.g., EMPLOYER). These tables are built from map queries residing on the client application (e.g., mEMPLOYER) and have the prefix "m". A second table contains records that have not yet been geocoded and are based on geocoding queries residing on the client application. These tables have the same name as their mapped table counterpart with the addition of the prefix "gc" (e.g., gcEMPLOYER), for geocoding. This is the table to geocode with, and the mapped table is the table that is added as an event theme and visible in the view (see figure 1.3).

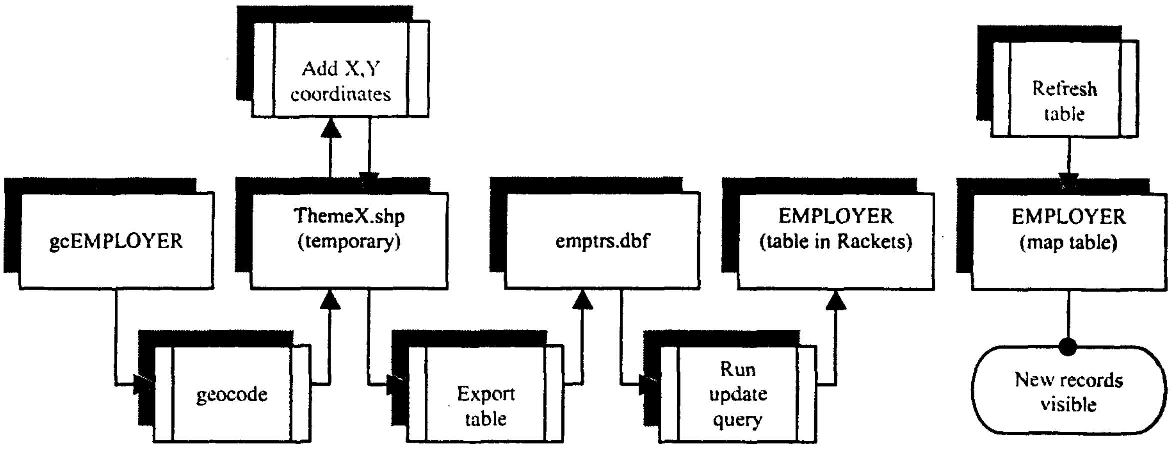


Figure 1.4

The process necessary to go from the geocode table to the virtual table that is added as an event theme is outlined in figure 1.4 for the Employer table. The steps have been automated by the script "Rackets.Geocode" which is run from the Rackets menu item "Geocode new records." First, the records are geocoded and a temporary shape file is created, then the x and y coordinates are generated from the shape field and the file is exported as a dbase file (emptrs.dbf). This file is linked to the Rackets database and an existing update query in Rackets is run via a macro using Dynamic Data Exchange (DDE) to insert the x and y coordinates into the corresponding records. Once the records are updated, the map table (e.g., EMPLOYER) can be refreshed to view the newly geocoded records.

There are eight such mapping tables in the application and others can be added following the same process. The above example outlines the procedure for the Employer table in Rackets. Table 1.1 outlines all the Rackets themes, tables and queries used in the geocoding and data visualization process.

Event theme	Map table	Geocode table	Update query	Macro
Residences	**DEFENDANTS**	**gcDEFENDANTS**	**uqDefXY**	**mcrDefXY**
Cooperation Area	**COOP_AREA**	**gcCOOPAREA**	**uqCoopAreaXY**	**mcrCoopAreaXY**
ARREST	**ARREST**	**gcARREST**	**uqArrXY**	**mcrArrXY**
STATUTES	**STATUTES**	**gcSTATUTES**	**uqStatuesXY**	**mcrStatutesXY**
EMPLOYER	**EMPLOYER**	**gcEMPLOYER**	**uqEmployXY**	**mcrEmployXY**
LOCALE	**LOCALE**	**gcLOCALE**	**uqLocaleXY**	**mcrLocaleXY**
COHORT	**COHORT**	**gcCOHORT**	**uqCohortXY**	**mcrCohortXY**
HOMICIDES	**HOMICIDES**	**gcHOMICIDES**	**uqHomicideXy**	**mcrHomicideXY**

Table 1.1

Base Maps

The customization of the mapping application also included the selection, purchase and setup of base data layers. All of the base maps were added to the project and their legends and properties were set (see figure 1.3). The base maps are Geographic Data Technologies' Dynamap 2000 for the five boroughs of New York City. The base maps include the following layers:

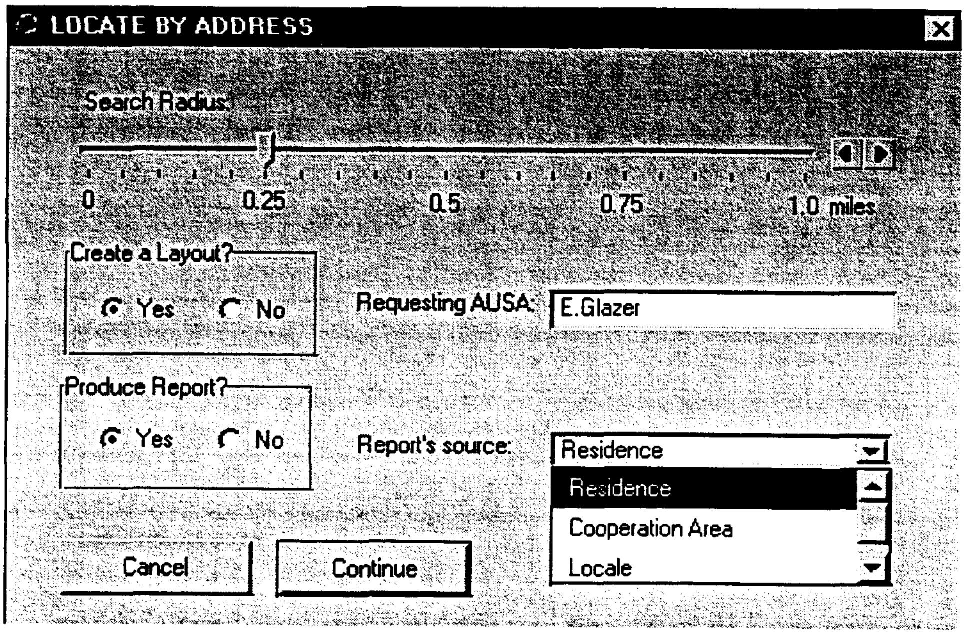
- Major streets
- Local streets
- Water bodies
- Parks
- Landmarks
- Institutions
- County polygons
- U.S. Census tracts
- U.S. Census Block groups

In addition, New York Police Department (NYPD) precinct and precinct census boundaries were created for the GDT base maps from NYPD maps and overlay directly on the streets.

To assist in the geocoding process a customized table of place name aliases (e.g., WTC for World Trade Center) was developed. ArcView uses this alias table to locate places with which it is unfamiliar. The table was developed over the course of the year as common aliases that were not being geocoded became evident.

Queries and Procedures

Avenue scripts were written to automate the most frequent queries run by the SDNY. The first such procedure, selects and zooms to specified records (e.g., Cooperation Area; an area given by a federal cooperator for which he\she has specific knowledge) within a certain radius of a specified address or intersection. Depending on the user input, the procedure produces a map layout and/or report for printing based on the search results. Information for the query is entered in the dialog below (figure 1.5) which is run from the Rackets main menu (see figure 1.3). Prior to showing this query dialog, a dialog prompts the user for an address. If the address is found the Locate By Address dialog appears, otherwise a message appears indicating that the address could not be located and notifies the user to check the address and re-enter it or try another address.


LOCATE BY ADDRESS

Search Radius: 0.25

Create a Layout? ● Yes ○ No

Requesting AUSA: E.Glazer

Produce Report? ● Yes ○ No

Report's source: Residence

Figure 1.5

If the user chooses to generate a report, the report gives detailed information contained in the database for the records selected within the search radius. For example, if an AUSA has a new case on Maple Ave & Smith St and she would like to know if there are any cooperators with knowledge of the area, she could run the above query selecting Cooperation Area as the Report's source and the procedure would produce a report listing all federal cooperators entered in Rackets and within the specified radius.

The next procedure Select By Sector selects all records from a specified table ("Select Records from:") contained within a selected NYPD precinct sector. This query uses the dialog below (figure 1.6) to gather the query information specified above from the user. A selection table is created containing the selected records for this query.

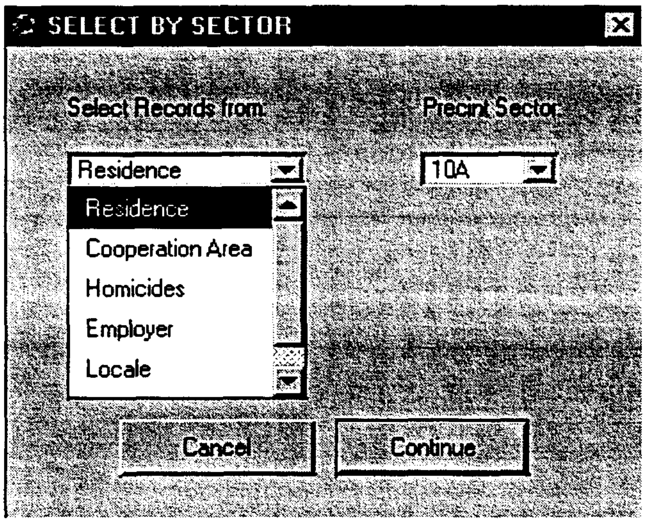


Figure 1.6

The final procedure from the Rackets menu, Incidence Map counts the number of events (e.g., Cooperators, Homicides) within each precinct sector and classifies the resulting theme by the number of events per sector. Figure 1.7 is an example of the output from this operation. This offers the Attorney's Office a simplified view of the spatial distribution of their docket at any given point in time.

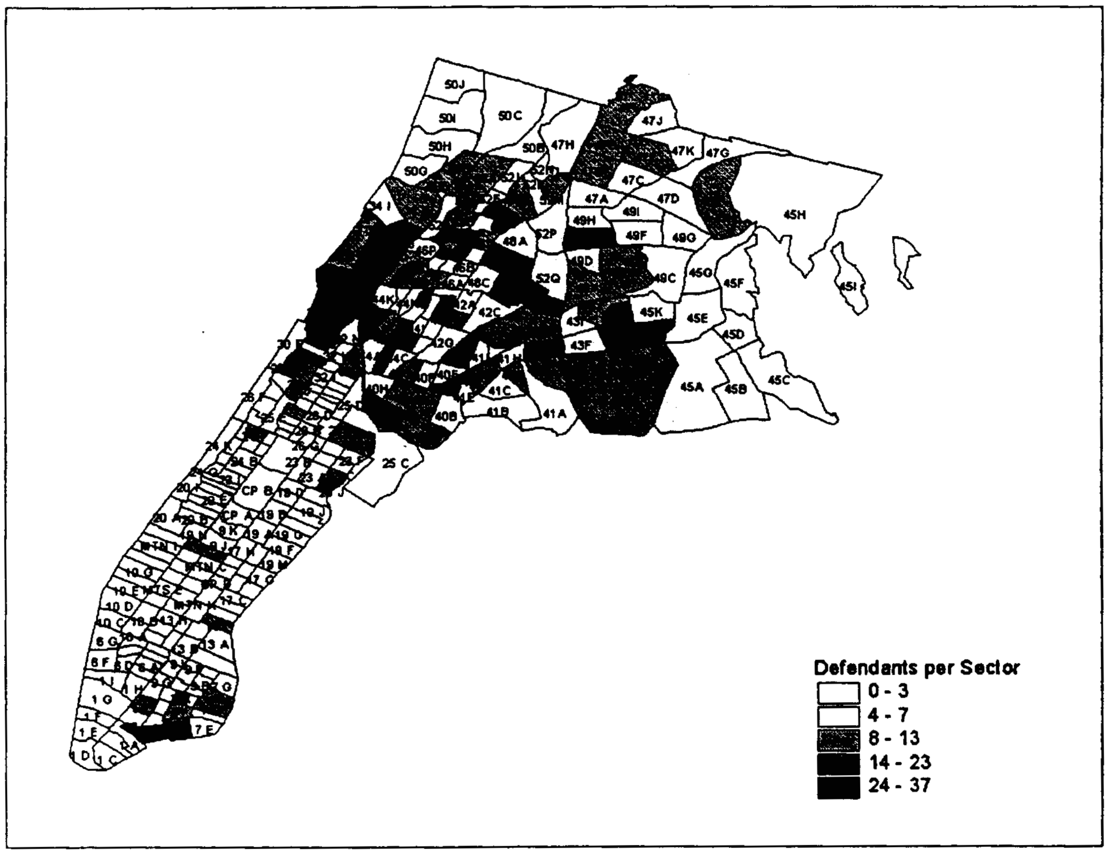


Figure 1.7

Conclusion

This manual provides a brief overview and description of Rackets. The development of Rackets has been an ongoing and evolving process over the last year. Many changes and updates were incorporated into the system over the course of the year. The evolution of the system was evident, as it went from running on a stand-alone PC, to a local area network with three workstations. Ideas were hatched and visions clarified as we moved along with the work. From a concept came a system and potentially a powerful tool as envisioned by its creator, Elizabeth Glazer. Hopefully the work will prove beneficial and further work will be done to improve and expand the system.

Appendix A- Data Dictionary

The data dictionary details every table and field included in the database. All queries and reports derived from the system are based on the information found within these tables. Users can update and navigate through this information through the client application / user interface.

Appendix A

Data Dictionary

Tables:

ALIAS

Field Name	Data Type	Field Size	Special	Description
AliasKey	Number (Long)	4	Primary key	Auto Number
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (1:M)
Alias	Text	50		Alias
AliasLN	Text	30		Alias Last name
Alias_rm	Text	30		Remarks

ARREST

Field Name	Data Type	Field Size	Special	Description
ArrKey	Number (Long)	4	Primary key	Auto Number
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (1:M)
ArrDate	Date/Time	8		Arrest date
Location	Text	50		Location of arrest (e.g., Restaurant, Home)
Address	Text	50		Address of arrest
City	Text	20		City of arrest
State	Text	2		State of arrest
Zip	Text	10		Zip code of arrest
Remarks	Text	70		Comments
Xcoord	Number (Single)	4		X coordinate – latitude
Ycoord	Number (Single)	4		Y coordinate – longitude
Av_status	Text	9		Geocoding status (M=matched, U=unmatched

CASE

Field Name	Data Type	Field Size	Special	Description
CaseKey	Number (Long)	4	Primary key	Auto Number
USAONum	Text	20		USAO number
IndictmentNum	Text	30		Instrument number
CaseName	Text	80		Title of case
AUSA	Text	70	LUT	Assistant US Attorney
Unit	Text	4	LUT	Prosecuting unit
agency_code	Text	7	LUT	Abbreviation of arresting agency
Agent	Text	40		Investigating agent
StatuteNum	Text	25	LUT	Statute number of lead charge
OpenDate	Date/Time	8		Date case opened
CloseDate	Date/Time	8		Date case closed
EnterDate	Date/Time	8		Date case entered into RACKETS
Comments	Memo	-		Comments

COOP_AREA

Field Name	Data Type	Field Size	Special	Description
CoopAreaKey	Number (Long)	4	Primary key	Auto Number
CoopKey	Number (Long)	4	Foreign key	
CoopArea	Text	70		Cooperation area
NypdSec	Text	8		NYPD sector
Topic	Text	20	LUT	Cooperation topic (e.g., narcotics, homicide)
City	Text	25		City of cooperation area
State	Text	2		State of cooperation area
Zip	Text	10		Zip code of cooperation area
Xcoord	Number (Single)	4		X coordinate – latitude
Ycoord	Number (Single)	4		Y coordinate – longitude
Av_status	Text	9		Geocoding status (M=matched, U=unmatched

COOP_PEOPLE

Field Name	Data Type	Field Size	Special	Description
CPKey	Number (Long)	4	Primary key	Auto Number
CoopKey	Number (Long)	4	Foreign key	
CoopAreaKey	Number (Long)	4	Foreign key	
LName	Text	40		Last name
Activity	Text	50	LUT	Activity at location
FName	Text	35		First name
Remarks	Text	100		Comments

COOPERATOR

Field Name	Data Type	Field Size	Special	Description
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (1:M)
CoopKey	Number (Long)	4	Primary key	
CoopDate	Date/Time	8		
Status	Text	6	COOP or PROF	Status (Cooperator or Proffer)
Topic	Text	20	LUT	Cooperation topic
Comment	Memo	-		Comments

DEFENDANT

Field Name	Data Type	Field Size	Special	Description
DefKey	Number (Long)	4	Primary key	Auto number
CaseKey	Number (Long)	4	Foreign key	Link to Case table (1:M)
pr_key	Number (Long)	4	Foreign key	Link to PTS data
DefName	Text	35		Last name
DefFName	Text	30		First name
DefMName	Text	25		Middle name
Address	Text	60		Address of residence
Apartment	Text	12		Apartment of residence
City	Text	40		City of residence
State	Text	5		State of residence
Zipcode	Text	10		Zip code of residence
USMSNum	Text	50		US Marshals Service number
FBINum	Text	20		FBI number
SSNum	Text	20		Social security number
NYSID	Text	20		New York State ID
LeadDef	Yes/No	1		Lead defendant?
Coop	Yes/No	1		Cooperator?
Weapon	Yes/No	1		Weapon seized?
CrimeType	Text	20	LUT	Type of Crime
CrimeRole	Text	70		Role in crime
Sealed	Yes/No	1		Sealed?
Fugitive	Yes/No	1		Fugitive?
LicNum	Text	30		License number
LicSt	Text	3		License state
DOB	Text	8		Date of birth
POBcity	Text	40		Place of birth city
POBstate	Text	50		Place of birth state
race_code	Text	50	LUT	Abbreviation for race
Citizenship	Text	12		Country of citizenship
AlienNum	Text	20		Alien number
CounselLN	Text	30		Defense Counsel's last name
CounselFN	Text	30		Defense Counsel's first name
TelNum	Text	25		Telephone number
Beeper	Text	15		Beeper number
Xcoord	Number (Double)	8		X coordinate – latitude
Ycoord	Number (Double)	8		Y coordinate – longitude
Av_status	Text	9		Geocoding status (M=matched, U=unmatched
Updated	Date/Time	8		Date record entered or transferred from PTS
MugshotA	OLE Object	-		Mug shot photo – front
MugshotB	OLE Object	-		Mug shot photo - side

EMPLOYER

Field Name	Data Type	Field Size	Special	Description
DefKey	Number (Long)	4		Link to Defendant table (M:1)
EmplName	Text	50		Name of employer
Occupation	Text	25		Occupation of employee
Address	Text	50		Address of employer
City	Text	20		City of employer
State	Text	2		State of employer
Zip	Text	10		Zip code of employer
Xcoord	Number (Single)	4		X coordinate – latitude
Ycoord	Number (Single)	4		Y coordinate – longitude

FIREARMS

Field Name	Data Type	Field Size	Special	Description
FireKey	Number (Long)	4	Primary key	Auto Number
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (1:M)
CoopKey	Number (Long)	4		
FirearmCode	Text	2		Type of firearm
Make	Text	3		Make of firearm
Model	Text	30		Model of firearm
Caliber	Text	4		Caliber of firearm
SerialNum	Text	20		Serial number
PurchaseDate	Date/Time	8		Purchase date
Trace	Yes/No	1		Was the firearm traced?
Source	Text	35		Source of firearm (e.g., store, individual)
Address	Text	50		Address of source
City	Text	25		City of source
State	Text	2		State of source
Zip	Text	10		Zip code of source
Remarks	Text	70		Comments
Xcoord	Number (Single)	4		X coordinate – latitude
Ycoord	Number (Single)	4		Y coordinate – longitude

GROUPS

Field Name	Data Type	Field Size	Special	Description
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (M:1)
GroupName	Text	50		Name of group/organization/enterprise
Location	Text	30		General Location of group
Xcoord	Number (Single)	4		X coordinate – latitude (NOT USED)
Ycoord	Number (Single)	4		Y coordinate – longitude (NOT USED)

HOMICIDE

Field Name	Data Type	Field Size	Special	Description
HomKey	Number (Long)	4	Primary key	Auto number
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (1:M)
CoopKey	Number (Long)	4		
Status	Text	15		Status of homicide
VicLName	Text	30		Victim's last name
VicFName	Text	25		Victim's first name
TgtLName	Text	35		Target's last name
TgtFName	Text	30		Target's first name
HomDate	Date/Time	8		Date of Homicide
Address	Text	50		Address of homicide
City	Text	20		City of homicide
State	Text	2		State of homicide
Zipcode	Text	10		Zip code of homicide
Xcoord	Number (Single)	4		X coordinate – latitude
Ycoord	Number (Single)	4		Y coordinate – longitude

LOCALE

Field Name	Data Type	Field Size	Special	Description
LocKey	Number (Long)	4	Primary key	Auto number
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (1:M)
LocName	Text	30		Name of locale
Address	Text	50		Address of locale
Apartment	Text	12		Apartment of locale
City	Text	40		City of locale
State	Text	5		State of locale
Zipcode	Text	10		Zip code of locale
Comment	Memo	-		Comments
Xcoord	Number (Single)	4		X coordinate – latitude
Ycoord	Number (Single)	4		Y coordinate – longitude
Av_status	Text	9		Geocoding status (M=matched, U=unmatched

MISC

Field Name	Data Type	Field Size	Special	Description
MiscKey	Number (Long)	4	Primary key	Auto number
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (1:M)
Item	Text	70		Miscellaneous item
Desc	Text	150		Description

NARCOTIC

Field Name	Data Type	Field Size	Special	Description
NarcKey	Number (Long)	4	Primary key	Auto number
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (1:M)
NarcType	Text	15	LUT	Type of narcotic
Brand	Text	30		Brand of Narcotic (i.e., street name)

Origin	Text	30		Origin of narcotic

REL_CASES

Field Name	Data Type	Field Size	Special	Description
CaseKey	Number (Long)	4	Foreign key	Link to Case table (M:1)
RelCase	Text	20		Related case's USAO number
Comments	Text	100		Comments

RELATIONS

Field Name	Data Type	Field Size	Special	Description
RelKey	Number (Long)	4	Primary key	Auto number
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (1:M)
rel_desc	Text	15	LUT	Type of relation (e.g., Brother, Mother)
RelLName	Text	30		Relation's last name
RelFName	Text	25		Realtion's first name
Address	Text	50		Address of relation
City	Text	20		Address of relation
State	Text	2		Address of relation
Zip	Text	10		Address of relation
Xcoord	Number (Single)	4		X coordinate – latitude
Ycoord	Number (Single)	4		Y coordinate – longitude

STATUTES

Field Name	Data Type	Field Size	Special	Description
StatKey	Number (Long)	4	Primary key	Auto number
DefKey	Number (Long)	4	Foreign key	Link to Defendant table (1:M)
Count	Text	15		Count
Supercede	Text	2		Superceding indictment
StatuteNum	Text	25	LUT	Statute number
Victim	Text	40		Victim of crime
Location	Text	40		Location of crime (e.g., restaurant)
Address	Text	50		Address of crime
City	Text	20		City of crime
State	Text	2		State of crime
Zip	Text	10		Zip code of crime
Xcoord	Number (Single)	4		X coordinate – latitude
Ycoord	Number (Single)	4		Y coordinate – longitude
Av_status	Text	9		Geocoding status (M=matched, U=unmatched

TARGET_ALIAS

Field Name	Data Type	Field Size	Special	Description
TgtAliasKey	Number (Long)	4	Primary key	Auto number
TargetKey	Number (Long)	4	Foreign key	Link to target table
TgtAlias	Text	35		Alias of target
TgtAliasLast	Text	40		Alias last name
TgtAliasRemark	Text	100		Comments

TARGETS

Field Name	Data Type	Field Size	Special	Description
TargetKey	Number (Long)	4	Primary key	Auto number
CaseKey	Number (Long)	4	Foreign key	Link to Case table (1:M)
TargetLast	Text	35		Last name of target
TargetFirst	Text	30		First name of target
TargetMiddle	Text	30		Middle name of target
Address	Text	60		Address of target
Apartment	Text	12		Apartment
City	Text	40		City
State	Text	5		State
Zipcode	Text	10		Zip code
SSNum	Text	20		Social security number
NYSID	Text	20		New York State Identification number
FBI	Text	15		FBI number
Photo	OLE Object	-		Photo
Remarks	Memo	-		Comments
Xcoord	Number (Double)	8		X coordinate – latitude
Ycoord	Number (Double)	8		Y coordinate – longitude
Av_status	Text	9		Geocoding status (M=matched, U=unmatched

WARRANT

Field Name	Data Type	Field Size	Special	Description
WarrantKey	Number (Long)	4	Primary key	Auto number
CaseKey	Number (Long)	4	Foreign key	Link to Case table (1:M)
SearchDate	Date/Time	8		Date of search
Address	Text	50		Address searched
Apartment	Text	12		Apartment
City	Text	20		City
State	Text	2		State
Zipcode	Text	10		Zip code
Xcoord	Number (Single)	4		X coordinate – latitude
Ycoord	Number (Single)	4		Y coordinate – longitude
Av_status	Text	9		Geocoding status (M=matched, U=unmatched

WARRANT_INVT

Field Name	Data Type	Field Size	Special	Description
WarrantKey	Number (Long)	4	Foreign key	Link to Warrant table (M:1)
Item	Text	50		Item confiscated

WARRANT_NAMES

Field Name	Data Type	Field Size	Special	Description
WarrantKey	Number (Long)	4	Foreign key	Link to Warrant table (M:1)
LName	Text	30		Last name on warrant
FName	Text	25		First name on warrant

WIRE_NAMES

Field Name	Data Type	Field Size	Special	Description
WireKey	Number (Long)	4	Foreign key	Link to Wiretap table (M:1)
LName	Text	30		Last name on wiretap
FName	Text	25		First name on wiretap

WIRETAP

Field Name	Data Type	Field Size	Special	Description
WireKey	**Number (Long)**	**4**	Primary key	Auto number
CaseKey	Number (Long)	4	Foreign key	Link to Case table (1:M)
Comments	Memo	-		Comments
TelNum	Text	50		Telephone number
Address	Text	50		Address of wiretap
Apartment	Text	12		Apartment
City	Text	20		City
State	Text	2		State
Zipcode	Text	10		Zip code
Xcoord	Number (Single)	4		X coordinate – latitude
Ycoord	Number (Single)	4		Y coordinate – longitude
Av_status	Text	9		Geocoding status (M=matched, U=unmatched

WIRETAP_DATES

Field Name	Data Type	Field Size	Special	Description
WireKey	Number (Long)	4	Foreign key	Link to Wiretap table (M:1)
BeginDate	Date/Time	8		Begin date
EndDate	Date/Time	8		End date

Lookup Tables

agency

Field Name	Data Type	Field Size	Special	Description
agency_code	Text	4		Agency code
agency_name	Text	35		Agency name

ausa

Field Name	Data Type	Field Size	Special	Description
AUSA	Text	50		Assistant US Attorney

coopTopic

Field Name	Data Type	Field Size	Special	Description
Topic	Text	50		Cooperation topic

crime_type

Field Name	Data Type	Field Size	Special	Description
CrimeType	Text	25		Crime type

coop_activity

Field Name	Data Type	Field Size	Special	Description
Activity	Text	50		Linked to COOP_AREA Topic

event_type

Field Name	Data Type	Field Size	Special	Description
event_type	Text	20		Type of event (e.g., Proffer, Indictment)

crimecode

Field Name	Data Type	Field Size	Special	Description
StatuteNum	Text	20		Statute number
StatDesc	Text	70		Description

narc_type

Field Name	Data Type	Field Size	Special	Description
NarcType	Text	20		Type of narcotic (e.g., heroin, cocaine)

unit

Field Name	Data Type	Field Size	Special	Description
Unit	Text	5		Unit code
UnitDesc	Text	30		Unit

***LUT: See Lookup table**

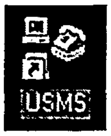
Appendix B -
Downloading PTS Data from USMS

Everyday, all new arrest information received from the United States Marshall's Service Prisoner Tracking System is transferred to the RACKETS system. Once downloaded these records are separated from the main database until they are properly transferred to the main database. Appendix B details the process by which data are initially brought into the RACKETS system from the United States Marshall's Service.

Instructions for downloading PTS data from USMS

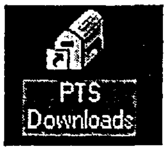
This must be done from RICHMOND.

1. Double-click USMS Dial-up Networking connection from desktop.

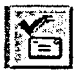


2. Enter password and hit Connect. When connection is established the dialog box will minimize to lower right corner of computer. *If there is a problem first check Dial-up Networking connection and then if the problem persists contact USMS.*

3. Double-click PTS Downloads.



4. Check for messages by clicking on the check mail icon. This should take a while depending on the number of downloads.



5. When completed, go to the Edit menu and choose Select All and hit delete.

6. Close Eudora (the email application) and disconnect form USMS by double clicking on the Dial-up Networking icon in the lower right of the computer and hitting Disconnect.

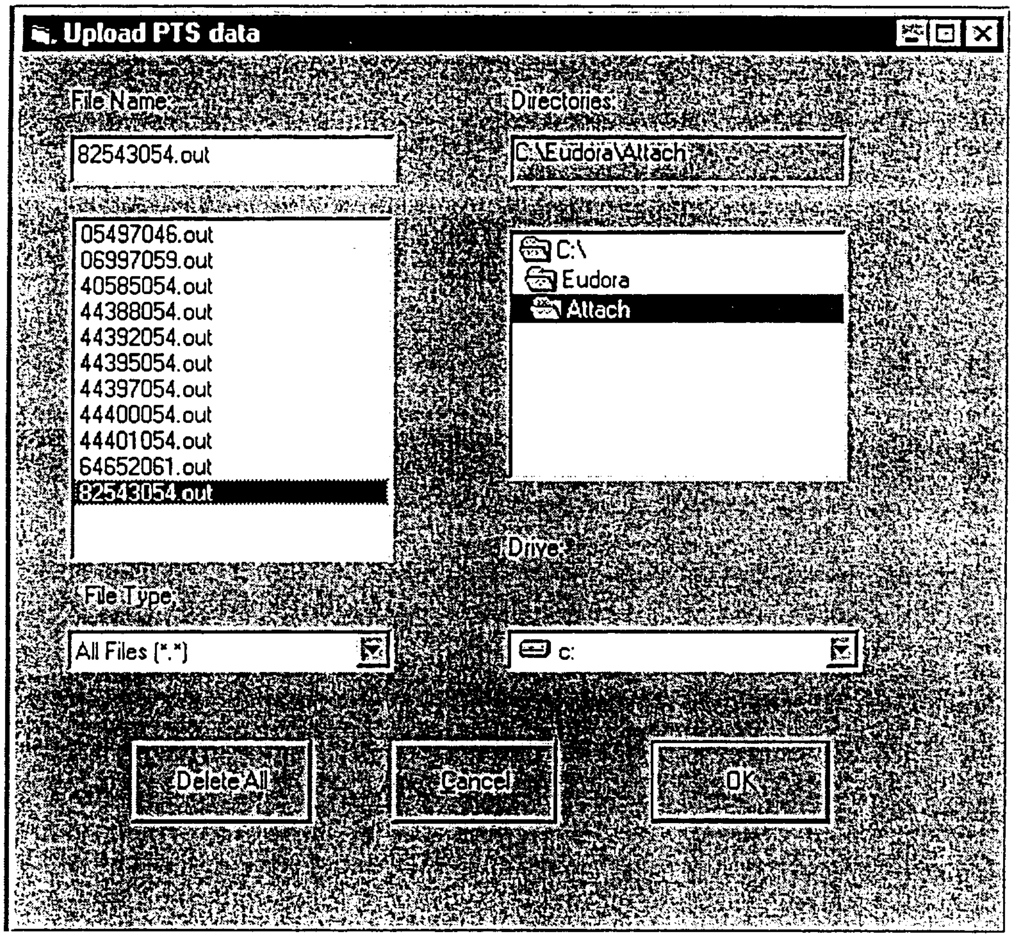
7. Open Windows Explorer from Start, Programs.

8. Navigate to C:\Eudora\Attach and sort to files by type by clicking on Type in the upper right portion of the screen.

9. Select all the image files (.jpg extension) by selecting the first, dolging down Shift and selecting the last image file. Then select Cut from the Edit menu and navigate to "Network on 'Bronx' Z:\mugs_t" and select Paste from Edit. Overwrite any old mug shots if prompted.

10. Open Rackets and go to the Maintenance screen.

11. Select Upload PTS emails transfers and navigate to C:\Eudora\Attach. Highlight the first message and hit OK, move to the next file and repeat, continuing until all messages have been transferred to the PTS database. Hit Cancel to Exit.



Upload PTS data

File Name: 82543054.out

```
05497046.out
06997059.out
40585054.out
44388054.out
44392054.out
44395054.out
44397054.out
44400054.out
44401054.out
64652061.out
82543054.out
```

Directories: C:\Eudora\Attach

```
C:\
  Eudora
    Attach
```

File Type: All Files (*.*)

Drive: c:

12. Navigate back to C:\Eudora\Attach and delete all files.

13. Back in Rackets, from the main menu select Reports to go to the main Report screen and click the report PTS Arrests (Date Range). From the next screen enter the preceding day as the begin and end dates and hit Print. These will be all the prisoners arrested the previous day and if the "Entered in Rackets" is blank the defendant has not yet been entered from the Rackets end and needs to be. Checked to see if they defendant is in Lions and proceed from there.

Appendix C - RACKETS Script

These scripts are written in Avenues, the programming language for Arcview (the RACKETS system GIS). These codes were specifically written to automate routine tasks within the mapping component of the RACKETS system. The first script, Rackets.LabelThemeUpdate, activates the label button when the scale on the map is less than or equal to 20,000. The Rackets.LabelTheme.Open script opens the label theme dialog box when the user presses the 'label theme' button. The label theme dialog box allows the user to determine what components of the map to label (such as streets, defendant names, agencies, cooperator, etc.) and then automatically labels what the user selects. The Rackets.LocateBy Address script selects all active records within a given radius of a specified address and produces a report and layout (when selected). The last script is used to produce a thematic map from a selected theme (Cooperation Area, Residence, Homicides, Arrest, Crimes) by summarizing the number of events per precinct sector.

Rackets.LabelThemeUpdate

```
'Colin Reilly
'2/98
'Update script for Rackets.LabelTheme
'Activates lable button if scale of view < 20,000

theView = av.GetActiveDoc
self.SetEnabled(FALSE)

'get scale
theScale = theView.ReturnScale

If (theScale.<=(20000)) Then
  self.SetEnabled(TRUE)
End
```

Rackets.LabelTheme.Open

```
'Colin Reilly
'2/98
'open script for the LabelTheme dialog

self.FindByName("lbtCancel").SetTag("FALSE")

chkStr = self.FindByName("chkStreets")
cpaStr = self.FindByName("cpaStrThm")
chkStr.SetListeners({cpaStr})

tList = {"Residence", "Cooperation Area",
        "Statutes", "Arrest"}
self.FindByName("cboRThm").DefineFromList(tList)

chkRkt = self.FindByName("chkRackets")
cboR = self.FindByName("cboRThm")
cpaL = self.FindByName("cpaLabel")
chkRkt.SetListeners({cboR,cpaL})

chkRkt.SetSelected(FALSE)
chkStr.SetSelected(TRUE)
```

Rackets.LocateByAdd

```
'Colin Reilly 1/98

'This script selects all active records w/in a given radius of a specified
address.
'Produces a report and layout when selected
'Dialog = LocateByAddress

thePrj = av.GetProject
theView = av.GetActiveDoc

'clear all existing selections and deactivate alll themes
for each t in theView.GetActiveThemes
  t.ClearSelection
  t.SetActive(FALSE)
end

'delete all existing graphics
theView.GetGraphics.Empty
tDisplay = theView.GetDisplay
theView.GetGraphics.Invalidate

'get matchtheme - theme is set to geocode using "US Streets"
strTheme = theView.FindTheme("NYC_geocode_s")
du = tDisplay.GetDistanceUnits
if (strTheme = nil) then
  exit
end
strTheme.SetActive(True)

'query for address to locate
pt = LocateDialog.Show(strTheme)

'if user hits cancel end
if (pt = NIL) then
  Return NIL
End

'if address is not found notify user and end
'get search address from dialog
LocAdd = LocateDialog.GetLastAddress
if (pt.IsNull) then
  MsgBox.Info(LocAdd++"was not found."
  +nl+"Make sure to use proper address format and try again."
  , "Cannot Locate Address")
  Return NIL
end

proj = theView.GetProjection
pp = pt.ReturnProjected(proj)
g = GraphicShape.Make(pp)

'query user for all necessary info
tDialog = av.FindDialog("LocateByAddress")
tDialog.Open
```

```
'If user hits cancel end script
If (tDialog.FindByName("lbtCancel").GetTag = "TRUE") then
  Return NIL
End

'get radius from dialog, if nil end
r = tDialog.FindByName("sldRadius").GetValue
if((r = nil) or (r = 0)) then
  Return NIL
end

'convert miles to decimal degrees
r2 = Units.Convert (r, #UNITS_LINEAR_MILES, #UNITS_LINEAR_DEGREES)

'make the circle
c = Circle.Make(pt,r2)
gc = GraphicShape.Make(c)
cSym = gc.GetSymbol
cSym.SetStyle(#RASTERFILL_STYLE_EMPTY)
cSym.SetOLColor(Color.GetBlack)
cSym.SetOLWidth(2)
tGList = theView.GetGraphics
theView.GetGraphics.Add(gc)
strTheme.SetActive(False)

'select features from active themes
thmList = {theView.FindTheme("RESIDENCE"),theView.FindTheme("STATUTES"),
          theView.FindTheme("ARREST"),theView.FindTheme("Cooperation Area")}


if (thmList.Count = 0) then
  msgBox.Warning("Themes were not found.","Script will end!")
  Return NIL
end
for each t in thmList
  t.SetActive(True)
  t.SelectByShapes({c},#VTAB_SELTYPE_NEW)
end
rb = gc.GetBounds
if (rb.IsEmpty) then
  return nil
elseif ( rb.ReturnSize = (0@0) ) then
  theView.GetDisplay.PanTo(rb.ReturnOrigin)
else
  theView.GetDisplay.SetExtent(rb.Scale(1.1))
end

'check dialog to produce a layout
go = tDialog.FindbyName("cpaLayout").GetSelected
If (go.AsString = "radLayoutYes") Then
  'creat a new layout
  newLayout = Layout.Make
  tempLst = TemplateMgr.GetTemplates
  'Get the layout template - "LocateAddress"
  '- referred to by its number in the list of templates
  t = tempLst.Get(5)
  if (t <> nil) then
```

```
    today = Date.Now.SetFormat("MMMM dd, yyyy").AsString
    txtLayout =
{LocAdd,r.AsString,tDialog.FindbyName("txlRequest").GetText,today}
    objs = newLayout.GetGraphics.GetFrameObjs
    objs.Merge(av.GetProject.GetDocs)
    objs.Merge(txtLayout)
    newLayout.UseTemplate(t)
    newLayout.GetGraphics.FillFrames(objs)
    newLayout.SetName(tDialog.FindbyName("txlRequest").GetText
    +Date.Now.SetFormat("MMddyy").AsString)
    lwin = newLayout.GetWin
    lwin.Open
    lwin.MoveTo(2,400)
    lwin.Resize(320,275)
  Else
  End
End

'check dialog to run report
cont = tDialog.FindByName("cpaReport").GetSelected

if (cont.AsString = "radReportYes") then
  theView.GetWin.Maximize
  'Generate reports
  rptTheme = tDialog.FindByName("cboReport").GetSelection
  vt = theView.FindTheme(rptTheme.AsString).GetFTab
  vt.Export("c:\temp\temp_rpt.dbf".AsFileName,dBASE,TRUE)
  ' Build dictionary to pass parameters to report
  rinDict = Dictionary.Make(5)
  rinDict.Add("RI_LIBRARY","d:\rackets_gis\reports\rackets.rp5")

  'run report based on theme selected
  If (rptTheme.AsString = "Residence") Then
    rpt = "residences"
  ElseIf (rptTheme.AsString = "Cooperation Area") Then
    rpt = "cooperators"
  End

  rinDict.Add("RI_REPORT", rpt)
  rinDict.Add("RI_TITLE",LocAdd)
  rinDict.Add("RI_PRINTER","D")
  rinDict.Add("RI_MASTER","c:\temp\temp_rpt.dbf")
  rinDict.Add("RI_STATFILE","c:\temp\rout.txt")
  'Call the main script, passing the dictionary object above
  av.run("PTRuntime.RunRpt",rinDict)
Else
end
```

```
'Colin Reilly 3/98
'generates a thematic map from the selected theme by summarizing the number of
'events per precinct sector

tView = av.GetActiveDoc
thmLst = {"Cooperation Area", "Residence","Homicides", "Arrest","Crimes"}
selThm = msgBox.ChoiceAsString(thmLst,"Select the theme to create a thematic map
from",
        "Thematic Map")
If (selThm = nil) then
  RETURN NIL
Else
End

'get selected theme
tTheme = tView.FindTheme(selThm.AsString)
toVTab = tTheme.GetFTab
toField = toVTab.FindField("Shape")

'get sector theme
frmThm = tView.FindTheme("bxmnsec")
fromThm = frmThm.Clone
fromVTab= fromThm.GetFTab
fromField = fromVTab.FindField("Shape")

toVTab.Join(toField,fromVTab,fromField)

secField = toVTab.FindField("Label")

'summarize vtab by sector
tDir = "c:\temp\".AsFileName
fName = tDir.MakeTmp("sum","dbf")
fVTab = toVTab.Summarize(fName, dbase, secField, {toField},
{#VTAB_SUMMARY_COUNT})

'make a copy of bxmansec to create thematic map from
fName2 = tDir.MakeTmp("thmmap","shp")
thmFTab = fromVTab.Export(fName2, Shape, FALSE)
sn = SrcName.Make(fName2.AsString)
newTheme = Theme.Make(sn)
newTheme.SetName (tTheme.GetName++"per Sector")
newVTab = newTheme.GetFTab
tFld = newVTab.FindField("Label")

fFld = fVTab.FindField("Label")
newVTab.Join(tFld, fVTab, fFld)   - joined summarized vtab to new sector theme

'add a new count field and return zero for those sectors with no data
shpFld = newVTab.FindField("shape")
countFld  = newVTab.FindField("Count")
idFld = newVTab.FindField("Id")
newVTab.SetEditable(TRUE)
cFld = Field.Make("Count2",#FIELD_BYTE,4,0)
newVTab.AddFields({cFld})

'if count field is null set value to 0
For each s in newVTab
```

```
  x = newVTab.ReturnValueNumber(idFld,s)
  z = newVTab.ReturnValueNumber(countFld,s)
  If (Not(x = 0) and (z.IsNull)) Then
    newVTab.SetValueNumber(cFld,s,0)
  ElseIf (Not(x = 0) and (Not(z.IsNull))) Then
    newVTab.SetValueNumber(cFld,s,z)
  End
End
'newVTab.SetEditable(FALSE)


'set legend
tLgnd = newTheme.GetLegend
tLgnd.SetLegendType(#LEGEND_TYPE_COLOR)
'use new count field for legend and divide into 5 classes
tLgnd.Natural(newTheme,"Count2",5)
'use - yellow to orange to red color scheme
theColorRamp = SymbolList.GetPreDefined(#SYMLIST_TYPE_COLORRAMP).Get(19)
tLgnd.GetSymbols.RampSavedColors(theColorRamp)
tLgnd.DisplayNoDataClass(FALSE)
'do not display no data class
newTheme.UpdateLegend

'add theme to view
tView.AddTheme ( newTheme )
NewTheme.SetVisible( TRUE )
NewTheme.SetActive(TRUE)
tView.Invalidate
toVTab.UnjoinAll

'zoom to bronx and manhattan
theThemes = tView.GetActiveThemes
r = Rect.MakeEmpty
for each t in theThemes
  r = r.UnionWith(t.ReturnExtent)
end

if (r.IsEmpty) then
  RETURN NIL
elseif ( r.ReturnSize = (0@0) ) then
  tView.GetDisplay.PanTo(r.ReturnOrigin)
else
  tView.GetDisplay.SetExtent(r.Scale(1.1))
end
```

Appendix D -

Microsoft Access Visual Basic Code (pg 1-65)

These codes create a user-friendly interface (Access forms) to facilitate data entry and querying in the underlying data tables. In general, the codes are used to search for individual records, to create new records, to close out records and open new ones, to select the appropriate entries under certain categories, to query the data, and to let the data entry person know if a defendant or case is already in the system. The underlying codes provide the basis for a relatively simplistic data entry format.

Form: CASE

Code

```
Attribute VB_Name = "Form_CASE"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit


Private Sub AUSA_NotInList(NewData As String, Response As Integer)

Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of AUSAs?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("ausa")
    rstTypes.AddNew
    rstTypes!AUSA = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If

End Sub

Sub Combo15_AfterUpdate()
    ' Find the record that matches the control.
    If IsNull(Me![Combo15]) Then
        Exit Sub
    End If

    Me.RecordsetClone.FindFirst "[CaseKey] = " & Me![Combo15]
    Me.Bookmark = Me.RecordsetClone.Bookmark
End Sub



Private Sub Command50_Click()

End Sub

Private Sub Combo15_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub Form_Current()
    Combo15 = CaseKey
End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
Private Sub cmdPrevious_Click()
On Error GoTo Err_cmdPrevious_Click


    DoCmd.GoToRecord , , acPrevious

Exit_cmdPrevious_Click:
    Exit Sub

Err_cmdPrevious_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrevious_Click

End Sub
Private Sub cmdNext_Click()
On Error GoTo Err_cmdNext_Click


    DoCmd.GoToRecord , , acNext

Exit_cmdNext_Click:
    Exit Sub

Err_cmdNext_Click:
    MsgBox Err.Description
    Resume Exit_cmdNext_Click

End Sub
Private Sub cmdAddRecord_Click()
```

```
On Error GoTo Err_cmdAddRecord_Click


    DoCmd.GoToRecord , , acNewRec
    Me!USAONum.SetFocus

Exit_cmdAddRecord_Click:
    Exit Sub

Err_cmdAddRecord_Click:
    MsgBox Err.Description
    Resume Exit_cmdAddRecord_Click

End Sub
Private Sub cmdArchive_Click()
On Error GoTo Err_cmdArchive_Click

    Dim stDocName As String
    Dim stDocName1 As String
    Dim stDocName2 As String
    Dim stDocName3 As String
    Dim stDocName4 As String
    Dim stDocName5 As String
    Dim stDocName6 As String

    stDocName = "arcCASE"
    stDocName1 = "arcDEFENDANT"
    stDocName2 = "arcALIAS"
    stDocName3 = "arcSTATUTES"
    stDocName4 = "arcEVENTS"
    stDocName5 = "arcGROUPS"
    stDocName6 = "delCASE"

    DoCmd.OpenQuery stDocName, acNormal, acEdit
    DoCmd.OpenQuery stDocName1, acNormal, acEdit
    DoCmd.OpenQuery stDocName2, acNormal, acEdit
    DoCmd.OpenQuery stDocName3, acNormal, acEdit
    DoCmd.OpenQuery stDocName4, acNormal, acEdit
    DoCmd.OpenQuery stDocName5, acNormal, acEdit
    DoCmd.OpenQuery stDocName6, acNormal, acEdit

Exit_cmdArchive_Click:
    Exit Sub

Err_cmdArchive_Click:
    MsgBox Err.Description
    Resume Exit_cmdArchive_Click

End Sub

Private Sub Form_Error(DataErr As Integer, Response As Integer)
```

```

Const conErrFieldRequired = 3314

If DataErr = conErrFieldRequired Then
    DisplayMessage "You must enter a Case name and USAO number to save the record."
    Response = acDataErrContinue
Else
    Response = acDataErrDisplay
End If

End Sub

Private Sub Form_Load()
End Sub
```

Form: COOP_AREA

Code

```
Attribute VB_Name = "Form_COOP_AREA"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Topic_NotInList(NewData As String, Response As Integer)
Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "' to the list of Activities?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("coop_activity")
    rstTypes.AddNew
    rstTypes!Activity = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If

End Sub
```

Form: COOP_PEOPLEsubArea

Code

```
Attribute VB_Name = "Form_COOP_PEOPLEsubArea"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit



Private Sub Form_BeforeUpdate(Cancel As Integer)
Me![CoopKey] = Me.Parent![CoopKey]
End Sub
```

Form: COOPERATORS

Code

```
Attribute VB_Name = "Form_COOPERATORS"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cboFindName_AfterUpdate()

Me.RecordsetClone.FindFirst "[DefKey] = " & Me![cboFindName]
Me.Bookmark = Me.RecordsetClone.Bookmark


End Sub

Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub

Private Sub LName_AfterUpdate()

'Fill in Last name from defendant table
strFilter = "Me!DefKey = Forms!DEFENDANT!DefKey"
Me!LName = DLookup("[DefLName]", "DEFENDANT", strFilter)

End Sub

Private Sub LName_BeforeUpdate(Cancel As Integer)

End Sub
Private Sub cmdArchive_Click()
On Error GoTo Err_cmdArchive_Click

    Dim stDocName As String
    Dim stDocName1 As String
    Dim stDocName2 As String
    Dim stDocName3 As String
    Dim stDocName4 As String
    Dim stDocName5 As String

    stDocName = "arcCOOPERATOR"
    stDocName1 = "arcCOOP_AREA"
    stDocName2 = "arcCOOP_GROUP"
    stDocName3 = "arcHOMICIDE"
    stDocName4 = "arcNARCOTIC"
    stDocName5 = "delCOOPERATOR"

    DoCmd.OpenQuery stDocName, acNormal, acEdit
    DoCmd.OpenQuery stDocName1, acNormal, acEdit
    DoCmd.OpenQuery stDocName2, acNormal, acEdit
    DoCmd.OpenQuery stDocName3, acNormal, acEdit
    DoCmd.OpenQuery stDocName4, acNormal, acEdit
    DoCmd.OpenQuery stDocName5, acNormal, acEdit
```

```

Exit_cmdArchive_Click:
    Exit Sub

Err_cmdArchive_Click:
    MsgBox Err.Description
    Resume Exit_cmdArchive_Click

End Sub

Private Sub Combo42_AfterUpdate()

End Sub

Private Sub Form_Current()

'Synchronize combobox and records
cboFindName = DefKey

End Sub
Private Sub cmdPrevious_Click()
On Error GoTo Err_cmdPrevious_Click


    DoCmd.GoToRecord , , acPrevious

Exit_cmdPrevious_Click:
    Exit Sub

Err_cmdPrevious_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrevious_Click

End Sub
Private Sub cmdNext_Click()
On Error GoTo Err_cmdNext_Click


    DoCmd.GoToRecord , , acNext

Exit_cmdNext_Click:
    Exit Sub

Err_cmdNext_Click:
    MsgBox Err.Description
    Resume Exit_cmdNext_Click

End Sub
Private Sub cmdFirst_Click()
On Error GoTo Err_cmdFirst_Click


    DoCmd.GoToRecord , , acFirst

Exit_cmdFirst_Click:
    Exit Sub

Err_cmdFirst_Click:
    MsgBox Err.Description
    Resume Exit_cmdFirst_Click

End Sub
Private Sub cmdLast_Click()
On Error GoTo Err_cmdLast_Click


    DoCmd.GoToRecord , , acLast

```

```
Exit_cmdLast_Click:
```

```
    Exit Sub

Err_cmdLast_Click:
    MsgBox Err.Description
    Resume Exit_cmdLast_Click

End Sub
Private Sub cmdPrntReport_Click()
On Error GoTo Err_cmdPrntReport_Click

    Dim stDocName As String

    stDocName = "COOPERATORS"
    DoCmd.OpenReport stDocName, acNormal

Exit_cmdPrntReport_Click:
    Exit Sub

Err_cmdPrntReport_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrntReport_Click

End Sub
```

Form: COOPERATORSsub

Code

```
Attribute VB_Name = "Form_COOPERATORSsub"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Topic_NotInList(NewData As String, Response As Integer)

'If item is not in list, check then add to list if requested
Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of Main Topics?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("coopTopic")
    rstTypes.AddNew
    rstTypes!Topic = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If

End Sub
```

Form: COOPERATORsubform

Code

```
Attribute VB_Name = "Form_COOPERATORsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdOpenCoopFrm_Click()
On Error GoTo Err_cmdOpenCoopFrm_Click

    'Save record first
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70

    Dim stDocName As String
    Dim stLinkCriteria As String
    'Dim stMsg As String
    'Dim stTitle As String

    'Check to see if there is a valid record selected before opening form
    'If IsNull(Me![CoopKey]) Then 'Notify user and then exit sub
    '    stMsg = "You must enter basic Cooperator information before you can select this option"
    '    stTitle = "Select a Record and then Continue"
    '        If MsgBox(stMsg, vbOKOnly, stTitle) = vbOK Then
    '            Exit Sub
    '        End If
    'End If

    stDocName = "COOPERATORS"

    stLinkCriteria = "[CoopKey]=" & Me![CoopKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenCoopFrm_Click:
    Exit Sub

Err_cmdOpenCoopFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCoopFrm_Click

End Sub

Private Sub Form_Current()

Me!CoopDate.SetFocus

End Sub

Private Sub Form_Unload(Cancel As Integer)

Me!CoopDate.SetFocus
Me.Repaint

End Sub
```

Form: CurrentPrisoners

Code

```
Attribute VB_Name = "Form_CurrentPrisoners"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Combo21_BeforeUpdate(Cancel As Integer)

End Sub


Private Sub Form_Current()

'Synchronize combobox with form
Combo21 = Me!FullName

End Sub


Private Sub Form_Load()

End Sub

Private Sub cmdImportRackets_Click()
On Error GoTo Err_cmdImportRackets_Click

'Confirm transfer
Dim strMsg As String
strMsg = "You are about to import a record from PTS to RACKETS, Continue?"
If MsgBox(strMsg, vbOKCancel, "Error!") = vbCancel Then
    Exit Sub
Else
End If

End Sub

Private Sub Form_Open(Cancel As Integer)

Me.pr_street.SetFocus

End Sub

Private Sub FormFooter_Click()

End Sub

Private Sub FormHeader_Click()

End Sub
Sub Combo21_AfterUpdate()
    ' Find the record that matches the control.
    Me.RecordsetClone.FindFirst "[FullName] = '" & Me![Combo21] & "'"
    Me.Bookmark = Me.RecordsetClone.Bookmark
End Sub


Private Sub pr_citizen_code_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub pr_citizen_code_Click()

End Sub
```

```
Private Sub cmdUpdateRec_Click()
On Error GoTo Err_cmdUpdateRec_Click

    Dim stDocName As String
    'Dim stDocName2 As String

    'Set dbs = CurrentDb
    'Set rst = dbs.OpenRecordset("DEFENDANT", dbOpenDynaset)

    stDocName = "qUpdateWPTS"
    'stDocName2 = "qApndArrest"

    DoCmd.OpenQuery stDocName, acNormal, acEdit

Exit_cmdUpdateRec_Click:
    Exit Sub

Err_cmdUpdateRec_Click:
    MsgBox Err.Description
    Resume Exit_cmdUpdateRec_Click

End Sub
Private Sub cmdNext_Click()
On Error GoTo Err_cmdNext_Click


    DoCmd.GoToRecord , , acNext

Exit_cmdNext_Click:
    Exit Sub

Err_cmdNext_Click:
    MsgBox Err.Description
    Resume Exit_cmdNext_Click

End Sub
Private Sub cmdPrevious_Click()
On Error GoTo Err_cmdPrevious_Click


    DoCmd.GoToRecord , , acPrevious

Exit_cmdPrevious_Click:
    Exit Sub

Err_cmdPrevious_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrevious_Click

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
Private Sub cmdArrest_Click()
```

```
On Error GoTo Err_cmdArrest_Click

    Dim stDocName As String

    stDocName = "qApndArrest"
    DoCmd.OpenQuery stDocName, acNormal, acEdit

Exit_cmdArrest_Click:
    Exit Sub

Err_cmdArrest_Click:
    MsgBox Err.Description
    Resume Exit_cmdArrest_Click

End Sub
```

Form: DEFENDANT

Code

```
Attribute VB_Name = "Form_DEFENDANT"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cboFindName_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub

Private Sub cmdSubmit_Click()

Dim rst As Recordset

    Dim strCriteria As String
    Dim strNYSID As String
    Dim strUSMS As String

        'Make sure there is an entry in the text box before running
        If IsNull(txtFindBy) Then
            MsgBox "Enter a NYSID or USMS numer and then submit query"
            Exit Sub
        End If

Select Case grpFindOptions


        Case 1 'Find case using NYSID number
            Set rst = Me.RecordsetClone
            strNYSID = Me!txtFindBy
            strCriteria = "[NYSID]='" & strNYSID & "'"
            rst.FindFirst strCriteria
            If rst.NoMatch Then
                MsgBox "The number you entered is not valid."
            Else
            Me.Bookmark = Me.RecordsetClone.Bookmark
            End If
            Me!txtFindBy.Value = ""
        Case 2 'Find case using USMS number
            Set rst = Me.RecordsetClone
            strUSMS = Me!txtFindBy
            strCriteria = "[USMSNum]='" & strUSMS & "'"
            rst.FindFirst strCriteria
            If rst.NoMatch Then
                MsgBox "The number you entered is not valid."
            Else
            Me.Bookmark = Me.RecordsetClone.Bookmark
            End If
            Me!txtFindBy.Value = ""
End Select

End Sub

Sub cboFindName_AfterUpdate()

    Dim rst As Recordset

    ' Find the record that matches the control.
    If IsNull(Me![cboFindName]) Then
        Exit Sub
    End If

    Set rst = Me.RecordsetClone
    rst.FindFirst "[DefKey] = " & Me![cboFindName]
    'Me.RecordsetClone.FindFirst "[DefKey] = " & Me![cboFindName]
    If rst.NoMatch Then
        rst.Close
        Exit Sub
    End If
    Me.Bookmark = rst.Bookmark

End Sub

Private Sub cmdNext_Click()
On Error GoTo Err_cmdNext_Click


    DoCmd.GoToRecord , , acNext

Exit_cmdNext_Click:
    Exit Sub

Err_cmdNext_Click:
    MsgBox Err.Description
    Resume Exit_cmdNext_Click

End Sub
Private Sub cmdPrevious_Click()
On Error GoTo Err_cmdPrevious_Click


    DoCmd.GoToRecord , , acPrevious

Exit_cmdPrevious_Click:
    Exit Sub

Err_cmdPrevious_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrevious_Click

End Sub

Private Sub Coop_AfterUpdate()

'If cooperator is true set cooperator page visible
If (Me!Coop = True) Then
    Me!tpgCooperator.Visible = True

ElseIf (Me!Coop = False) Then
    Me!tpgCooperator.Visible = False
Else
End If

End Sub

Private Sub Coop_BeforeUpdate(Cancel As Integer)
```

```


End Sub

Private Sub Coop_Click()

End Sub

Private Sub CrimeType_NotInList(NewData As String, Response As Integer)

Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of Crime Types?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("crime_type")
    rstTypes.AddNew
    rstTypes!CrimeType = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If

End Sub

Private Sub Detail_Click()

End Sub

Private Sub Form_AfterUpdate()

'Enable or disable mugshots button
If IsNull(Me!pr_key) Or IsNull(Me!txtMugshot) Then
    Me!cmdPhoto.Enabled = False
Else
    Me!cmdPhoto.Enabled = True
End If

'Enable or disable insert mugshots button
If IsNull(Me!txtMugshot) And Not (IsNull(Me!pr_key)) Then
    Me!cmdEnterPhoto.Enabled = True
Else
    Me!cmdEnterPhoto.Enabled = False
End If

End Sub

Private Sub Form_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub Form_Current()

'Set Find Record to current
cboFindName = DefKey

'If cooperator is true set cooperator page visible
If (Me!Coop = True) Then
    Me!tpgCooperator.Visible = True

ElseIf (Me!Coop = False) Then
```

```
    Me!tpgCooperator.Visible = False
Else
End If

'If Weapon seized is true set cooperator page visible
If (Me!Weapon = True) Then
    Me!tpgFIREARMS.Visible = True

ElseIf (Me!Weapon = False) Then
    Me!tpgFIREARMS.Visible = False
Else
End If

'Enable or disable mugshots button
If IsNull(Me!pr_key) Or IsNull(Me!txtMugshot) Then
    Me!cmdPhoto.Enabled = False
Else
    Me!cmdPhoto.Enabled = True
End If

'Enable or disable insert mugshots button
If IsNull(Me!txtMugshot) And Not (IsNull(Me!pr_key)) Then
    Me!cmdEnterPhoto.Enabled = True
Else
    Me!cmdEnterPhoto.Enabled = False
End If

End Sub
Private Sub cmdOpenCooperatorFrm_Click()
On Error GoTo Err_cmdOpenCooperatorFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "COOPERATORS"

    stLinkCriteria = "[DefKey]=" & Me![DefKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenCooperatorFrm_Click:
    Exit Sub

Err_cmdOpenCooperatorFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCooperatorFrm_Click

End Sub
Private Sub cmdOpenCooperatorForm_Click()
On Error GoTo Err_cmdOpenCooperatorForm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "COOPERATORS"

    stLinkCriteria = "[DefKey]=" & Me!DefKey
    DoCmd.OpenForm stDocName, , , stLinkCriteria, acAdd, , Me!DefKey


Exit_cmdOpenCooperatorForm_Click:
    Exit Sub

Err_cmdOpenCooperatorForm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCooperatorForm_Click

```

```
End Sub
```

Form: DEFENDANT

```

Private Sub Form_Error(DataErr As Integer, Response As Integer)
    Debug.Print "DataErr = "; DataErr
End Sub


Private Sub Frame99_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub Frame112_BeforeUpdate(Cancel As Integer)

End Sub


Private Sub grpFindOptions_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub LeadDef_AfterUpdate()

End Sub

Private Sub LeadDef_Click()

End Sub

Private Sub txtFindBy_BeforeUpdate(Cancel As Integer)

End Sub
Private Sub cmdOpenCaseFrm_Click()
On Error GoTo Err_cmdOpenCaseFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "CASE"
    'stDocName1 = "DEFENDANT"

    stLinkCriteria = "[CaseKey]=" & Me![CaseKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria


    'Close form
    DoCmd.Close acForm, "DEFENDANT", acSaveYes


Exit_cmdOpenCaseFrm_Click:
    Exit Sub

Err_cmdOpenCaseFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCaseFrm_Click

End Sub

Private Sub Weapon_AfterUpdate()

'If Weapon seized is true set cooperator page visible
If (Me!Weapon = True) Then
    Me!tpgFIREARMS.Visible = True

ElseIf (Me!Weapon = False) Then
    Me!tpgFIREARMS.Visible = False
Else
End If
End Sub

Private Sub Weapon_Click()

End Sub
Private Sub cmdPrintRpt_Click()
On Error GoTo Err_cmdPrintRpt_Click

    Dim stDocName As String

    stDocName = "DEFENDANT"
    DoCmd.OpenReport stDocName, acNormal

Exit_cmdPrintRpt_Click:
    Exit Sub

Err_cmdPrintRpt_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrintRpt_Click

End Sub
Private Sub cmdPhoto_Click()
On Error GoTo Err_cmdPhoto_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmDEF_PHOTOS"

    stLinkCriteria = "[DefKey]=" & Me![DefKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

    'do not allow records to be edited
    Forms![frmDEF_PHOTOS].oleMugA.Locked = True
    Forms![frmDEF_PHOTOS].oleMugB.Locked = True

Exit_cmdPhoto_Click:
    Exit Sub

Err_cmdPhoto_Click:
    MsgBox Err.Description
    Resume Exit_cmdPhoto_Click

End Sub
Private Sub cmdEnterPhoto_Click()
On Error GoTo Err_cmdEnterPhoto_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmDEF_PHOTOS"

    stLinkCriteria = "[DefKey]=" & Me![DefKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

    'Allow records to be edited
    Forms![frmDEF_PHOTOS].oleMugA.Locked = False
    Forms![frmDEF_PHOTOS].oleMugB.Locked = False

Exit_cmdEnterPhoto_Click:
    Exit Sub

Err_cmdEnterPhoto_Click:
    MsgBox Err.Description
    Resume Exit_cmdEnterPhoto_Click

```

```
400  End Sub
```

Form: DEFENDANTList

Code

```
Attribute VB_Name = "Form_DEFENDANTList"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit


Private Sub Form_Current()

'Enable or disable Details button
If IsNull(Me![DefKey]) Then
    Me!cmdOpenDefFrm.Enabled = False
Else
    Me!cmdOpenDefFrm.Enabled = True
End If

'Enable or disable PTS button
If IsNull(Me!USMSNum) Then
    Me!cmdSearchPTS.Enabled = True
Else
    Me!cmdSearchPTS.Enabled = False
End If
End Sub

Private Sub Form_Load()

'Enable or disable Details button
If IsNull(Me![DefKey]) Then
    Me!cmdOpenDefFrm.Enabled = False
Else
    Me!cmdOpenDefFrm.Enabled = True
End If

End Sub

Private Sub Form_Open(Cancel As Integer)

End Sub
Private Sub cmdOpenDefFrm_Click()
On Error GoTo Err_cmdOpenDefFrm_Click


    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim stTitle As String
    Dim stMsg As String
    Dim stMsgTrans As String
    Dim stTitleTrans As String
    Dim IndctNum As Variant
    Dim TopChg As Variant

    stDocName = "DEFENDANT"


    stLinkCriteria = "[DefKey]=" & Me![DefKey]

    'Save record first
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70

    'Check to see if there is a valid record selected before opening form
    'If IsNull(Me![DefKey]) Then 'Notify user and then exit sub
    '     stMsg = "You must be at a valid record before you can select this option"
    '    stTitle = "Select a Record and then Continue"
    '        If MsgBox(stMsg, vbOKOnly, stTitle) = vbOK Then
    '            Exit Sub
    '        End If
    'End If

    'get incictment# and top charge for use in transfer
    IndctNum = Forms![CASE]!IndictmentNum.Value
    TopChg = Forms![CASE]!StatuteNum.Value


    DoCmd.OpenForm stDocName, , , stLinkCriteria

    'Set focus to Middle Name
    Forms![Defendant]!DefMName.SetFocus

    'Disable all FindBy options
    Forms![Defendant]!grpFindOptions.Enabled = False
    Forms![Defendant]!cboFindName.Enabled = False
    Forms![Defendant]!cmdSubmit.Enabled = False
    Forms![Defendant]!txtFindBy.Enabled = False
    'Remove navigation buttons
    Forms![Defendant]!cmdPrevious.Visible = False
    Forms![Defendant]!cmdNext.Visible = False

    'Check if Case Instrument# & TopCharge should be entered for defendant
    'If IsNull(Forms!Defendant!tpgStatutes.Form.Controls!Count) And IsNull(Forms!Defendant!tpgStatutes.Form.Controls!StatuteNum) Then
    '    stMsgTrans = "Enter Case Instrument no. " & IndctNum & " and Top Charge " & TopChg & " for Defendant?"
    '    stTitleTrans = "RACKETS"
    '    If MsgBox(stMsgTrans, vbYesNo, stTitleTrans) = vbYes Then
    '        Forms![Defendant]![STATUTESsubform].Count.Value = 1
    '        Forms![Defendant]![STATUTESsubform].StatuteNum.Value = TopChg
    '    Else
    '    End If
    'End If

Exit_cmdOpenDefFrm_Click:
    Exit Sub

Err_cmdOpenDefFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenDefFrm_Click

End Sub
Private Sub cmdSearchPTS_Click()
On Error GoTo Err_cmdSearchPTS_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "CurrentPrisoners"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdSearchPTS_Click:
    Exit Sub

Err_cmdSearchPTS_Click:
    MsgBox Err.Description
    Resume Exit_cmdSearchPTS_Click

End Sub
```

Form: EVENTsubform

Code

```
Attribute VB_Name = "Form_EVENTsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit



Private Sub Form_AfterInsert()

Dim strMessage As String
Dim strLinkCriteria As String

strMessage = "Enter Proffer information now? At the very least, a topic must be entered to continue."

If Me.event_type = "Proffer" Then
        If Forms![Defendant].Coop.Value = False Then
            Forms![Defendant].Coop.Value = True
            Forms![Defendant]!tpgCooperator.Visible = True
        End If
        Forms![Defendant]!subCooperator.Form.Status = "PROF"
    'If Confirm(strMessage) Then
        Forms![Defendant]!tpgCooperator.SetFocus
        Forms![Defendant]!subCooperator.Form.Topic.SetFocus
    'Else
    'End If
End If

End Sub
```

Form: FIREARMSsubform

Code

```
Attribute VB_Name = "Form_FIREARMSsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub FirearmType_NotInList(NewData As String, Response As Integer)

Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of Firearm Types?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("firearm_type")
    rstTypes.AddNew
    rstTypes!FirearmType = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If


End Sub

Private Sub Make_NotInList(NewData As String, Response As Integer)

Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of Firearm Makes?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("firearm_make")
    rstTypes.AddNew
    rstTypes!Make = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If


End Sub

Private Sub Model_NotInList(NewData As String, Response As Integer)

Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of Firearm Models?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("firearm_model")
    rstTypes.AddNew
    rstTypes!Model = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If


End Sub
```

Form: frmDEF_PHOTOS

Code

```
Attribute VB_Name = "Form_frmDEF_PHOTOS"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click


    DoCmd.PrintOut

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click

    'save record first
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70

    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
```

Form: frmDefsByAUSA

Code

```
Attribute VB_Name = "Form_frmDefsByAUSA"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdPreview_Click()
On Error GoTo Err_cmdPreview_Click

    Dim stDocName As String

    stDocName = "rptDefsByAUSA"
    DoCmd.OpenReport stDocName, acPreview
    DoCmd.Close acForm, "frmDefsByAUSA"

Exit_cmdPreview_Click:
    Exit Sub

Err_cmdPreview_Click:
    MsgBox Err.Description
    Resume Exit_cmdPreview_Click

End Sub
Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click

    Dim stDocName As String

    stDocName = "rptDefsByAUSA"
    DoCmd.OpenReport stDocName, acNormal
    DoCmd.Close acForm, "frmDefsByAUSA"

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub
Private Sub cmdCancel_Click()
On Error GoTo Err_cmdCancel_Click


    DoCmd.Close

Exit_cmdCancel_Click:
    Exit Sub

Err_cmdCancel_Click:
    MsgBox Err.Description
    Resume Exit_cmdCancel_Click

End Sub
```

Form: frmDefsByUnit

Code

```
Attribute VB_Name = "Form_frmDefsByUnit"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdPreview_Click()
On Error GoTo Err_cmdPreview_Click

    Dim stDocName As String

    stDocName = "rptDefsByUnit"
    DoCmd.OpenReport stDocName, acPreview
    DoCmd.Close acForm, "frmDefsByUnit"

Exit_cmdPreview_Click:
    Exit Sub

Err_cmdPreview_Click:
    MsgBox Err.Description
    Resume Exit_cmdPreview_Click

End Sub
Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click

    Dim stDocName As String

    stDocName = "rptDefsByUnit"
    DoCmd.OpenReport stDocName, acNormal
    DoCmd.Close acForm, "frmDefsByUnit"

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub
Private Sub cmdCancel_Click()
On Error GoTo Err_cmdCancel_Click


    DoCmd.Close

Exit_cmdCancel_Click:
    Exit Sub

Err_cmdCancel_Click:
    MsgBox Err.Description
    Resume Exit_cmdCancel_Click

End Sub
```

Form: frmFindAlias

Code

```
Attribute VB_Name = "Form_frmFindAlias"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClear_Click()

Dim strSql As String
Dim strFullSql As String

Me!txtAlias = ""

strSql = "WHERE FALSE"
strFullSql = "Select * From qFindAlias " & strSql
Me!subFindAlias.Form.RecordSource = strFullSql

strFullSql = "Select • From qFindAKA " & strSql
Me!subFindAKA.Form.RecordSource = strFullSql

End Sub

Private Sub cmdSubmit_Click()

Dim strAlias As String
Dim strAliasSql As String
Dim strFullSql As String
Dim strMsg As String
Dim strTitle As String
Dim strWhere As String

If IsNull(Me!txtAlias) Or Me!txtAlias = "" Then
    strMsg = "Query is empty. Enter an alias and try again."
    strTitle = "Empty Query!"
        If MsgBox(strMsg, vbOKOnly, strTitle) = vbOK Then
           Exit Sub
        End If
End If

strWhere = "Where "

Select Case grpDatabase
'Search PTS
Case 1
    If grpMatch.Value = 1 Then
        strAlias = "([name_fname] = " & "Forms!frmFindAlias![txtAlias]OR[name_lname] =
" & "Forms!frmFindAlias![txtAlias]OR[name_mname] = " &
"Forms!frmFindAlias![txtAlias]OR[rm_desc] = " & "Forms!frmFindAlias![txtAlias])"

        strAliasSql = strWhere & strAlias
        strFullSql = "Select * From qFindAKA " & strAliasSql
        Me!subFindAKA.Form.RecordSource = strFullSql
    Else
        strAliasSql = " Where [name_fname] Like '" & Forms!frmFindAlias![txtAlias] &
"•' or [name_lname] Like '" & Forms!frmFindAlias![txtAlias] & "*' or [name_mname] Like
'" & Forms!frmFindAlias![txtAlias] & "*' or [rm_desc] Like '*" &
Forms!frmFindAlias![txtAlias] & "•'"

        strFullSql = "Select • From qFindAKA " & strAliasSql
        Me!subFindAKA.Form.RecordSource = strFullSql
    End If

```

```
'Search Rackets
```

```
Case 2
    If grpMatch.Value = 1 Then
        strAlias = "([Alias] = " & "Forms!frmFindAlias![txtAlias]OR[AliasLN] = " & "Forms!frmFindAlias![txtAlias])"
        strAliasSql = strWhere & strAlias
        strFullSql = "Select * From qFindAlias " & strAliasSql
        Me!subFindAlias.Form.RecordSource = strFullSql
    Else
        strAliasSql = " Where [Alias] Like '" & Forms!frmFindAlias![txtAlias] & "*' or [AliasLN] Like '" & Forms!frmFindAlias![txtAlias] & "*'"
        strFullSql = "Select • From qFindAlias " & strAliasSql
        Me!subFindAlias.Form.RecordSource = strFullSql
    End If
End Select

End Sub

Private Sub Form_AfterUpdate()

End Sub

Private Sub Form_Load()

End Sub

Private Sub Frame4_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub grpDatabase_AfterUpdate()

'Set subform visible based on selection of search
Select Case grpDatabase

Case 1
Me!subFindAKA.Form.Visible = True
Me!subFindAlias.Form.Visible = False

Case 2
Me!subFindAKA.Form.Visible = False
Me!subFindAlias.Form.Visible = True

End Select

End Sub

Private Sub txtAlias_BeforeUpdate(Cancel As Integer)

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
Private Sub cmdOpenDefFrm_Click()
On Error GoTo Err_cmdOpenDefFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

```

```
    'stLinkCriteria = Me!subFindAKA.Form.

    stDocName = "DEFENDANT"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenDefFrm_Click:
    Exit Sub

Err_cmdOpenDefFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenDefFrm_Click

End Sub
```

Form: frmFindAssoc

Code

```
Attribute VB_Name = "Form_frmFindAssoc"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClear_Click()
Dim strSql As String
Dim strFull As String

Me!txtFind = ""

strSql = "Where False"
strFull = "Select * From qFindAssoc " & strSql
Me!subFindAssoc.Form.RecordSource = strFull
Me!subFindLocale.Form.RecordSource = strFull
Me!subFindCohort.Form.RecordSource = strFull

End Sub

Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub

Private Sub cmdSubmit_Click()


Dim strAssoc As String
Dim strSql As String
Dim strMsg As String
Dim strTitle As String

'clear variables
'strAssoc = ""
'strSql = ""

'First check to see if a string was entered, if not end routine
strMsg = "Query is empty. Enter a Search String and try again"
strTitle = "RACKETS"
If IsNull(Me!txtFind) Or Me!txtFind = "" Then
    If MsgBox(strMsg, vbOKOnly, strTitle) = vbOK Then
        Exit Sub
    End If
End If

'run query based on selection type
Select Case grpType

'search Group table via qFindAssoc
Case 1
    strAssoc = "Where [GroupName] Like '" & Forms!frmFindAssoc![txtFind] & "*'"
    strSql = "Select * From qFindAssoc " & strAssoc
    Me!subFindAssoc.Form.RecordSource = strSql

'search locale via q FindLocale
Case 2

    strAssoc = "Where [lo_name] Like '" & Forms!frmFindAssoc![txtFind] & "*'"
    strSql = "Select * From qFindLocale " & strAssoc
    Me!subFindLocale.Form.RecordSource = strSql

 'search cohort via qFindCohort
Case 3
    strAssoc = "Where [co_lname] Like '" & Forms!frmFindAssoc![txtFind] & "*' or [co_lname] Like '" & Forms!frmFindAssoc![txtFind] & "*'"
    strSql = "Select • From qFindCohort " & strAssoc
    Me!subFindCohort.Form.RecordSource = strSql

End Select
End Sub

Private Sub Form_Close()
Me!txtFind = ""
End Sub

Private Sub Form_Open(Cancel As Integer)

Dim strSql As String
Dim strFull As String

'Clear last query
Me!txtFind = ""

strSql = "Where False"
strFull = "Select • From qFindAssoc " & strSql

Me!subFindAssoc.Form.RecordSource = strFull

'Set default visible and all others not visible
Me!subFindAssoc.Form.Visible = True
Me!subFindCohort.Form.Visible = False
Me!subFindLocale.Form.Visible = False


End Sub

Private Sub grpType_AfterUpdate()

'Set the subform based on the query type
Select Case grpType

Case 1
    Me!subFindAssoc.Form.Visible = True
    Me!subFindLocale.Form.Visible = False
    Me!subFindCohort.Form.Visible = False

Case 2
    Me!subFindLocale.Form.Visible = True
    Me!subFindAssoc.Form.Visible = False
    Me!subFindCohort.Form.Visible = False

Case 3
    Me!subFindCohort.Form.Visible = True
    Me!subFindAssoc.Form.Visible = False
    Me!subFindLocale.Form.Visible = False

End Select

End Sub
```

Form: frmFindCounsel

Code

```
Attribute VB_Name = "Form_frmFindCounsel"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cboFind_AfterUpdate()

Dim strSql As String
Dim strFull As String

strSql = "Select • From qFindCounsel Where [Counsel] = " &
"Forms!frmFindCounsel![cboFind]"
Me!subFindCounsel.Form.RecordSource = strSql


End Sub

Private Sub cboFind_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub cboFind_Change()
Dim strSql As String
Dim strFull As String

'strSql = "Select • From qFindCounsel Where [] = " & "Forms!frmFindCounsel![cboFind]"

'Me!subFindCounsel.Form.RecordSource = strSql


End Sub

Private Sub Detail_Click()

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:

    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
```

Form: frmFindEmployer

Code

```
Attribute VB_Name = "Form_frmFindEmployer"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClear_Click()

Dim strSql As String
Dim strFull As String

Me!txtFind = ""

strSql = "Where False"
strFull = "Select * From qFindEmployer " & strSql
Me!subFindEmployer.Form.RecordSource = strFull
Me!subFindEmployerPTS.Form.RecordSource = strFull

End Sub

Private Sub cmdSubmit_Click()

Dim strEmploy As String
Dim strSql As String
Dim strMsg As String
Dim strTitle As String


'First check to see if a string was entered, if not end routine
strMsg = "Query is empty. Enter a Search String and try again"
strTitle = "RACKETS"
If IsNull(Me!txtFind) Or Me!txtFind = "" Then
    If MsgBox(strMsg, vbOKOnly, strTitle) = vbOK Then
        Exit Sub
    End If
End If

'run query based on selection type
Select Case grpType

'search by employer
Case 1

    If grpSearch.Value = 1 Then
        strEmploy = "Where [em_name] Like '" & Forms!frmFindEmployer![txtFind] & "*'"
        strSql = "Select * From qFindEmployer " & strEmploy
        Me!subFindEmployer.Form.RecordSource = strSql
    Else
        strEmploy = "Where [em_name] Like '" & Forms!frmFindEmployer![txtFind] & "*'"
        strSql = "Select • From qFindEmployPTS " & strEmploy
        Me!subFindEmployerPTS.Form.RecordSource = strSql
    End If

'search by owner/supervisor
Case 2
    If grpSearch.Value = 1 Then
        strEmploy = "Where [em_poc] Like '*" & Forms!frmFindEmployer![txtFind] & "*'"
        strSql = "Select • From qFindEmployer " & strEmploy
        Me!subFindEmployer.Form.RecordSource = strSql
    Else
        strEmploy = "Where [em_poc] Like '*" & Forms!frmFindEmployer![txtFind] & "*'"
        strSql = "Select • From qFindEmployPTS " & strEmploy
        Me!subFindEmployerPTS.Form.RecordSource = strSql
    End If

'search by occupation
Case 3
    If grpSearch.Value = 1 Then
        strEmploy = "Where [em_occ] Like '" & Forms!frmFindEmployer![txtFind] & "*'"

        strSql = "Select • From qFindEmployer " & strEmploy
        Me!subFindEmployer.Form.RecordSource = strSql
    Else
        strEmploy = "Where [em_occ] Like '" & Forms!frmFindEmployer![txtFind] & "•'"

        strSql = "Select * From qFindEmployPTS " & strEmploy
        Me!subFindEmployerPTS.Form.RecordSource = strSql
    End If

End Select


End Sub

Private Sub Form_Open(Cancel As Integer)

Me!subFindEmployer.Form.Visible = True
Me!subFindEmployerPTS.Form.Visible = False

End Sub

Private Sub grpSearch_AfterUpdate()

'Set subform based on search selection
Select Case grpSearch

Case 1
'Search records linked to defendants in Rackets
Me!subFindEmployer.Form.Visible = True
Me!subFindEmployerPTS.Form.Visible = False

Case 2
'Search records not linked to an active defendant
Me!subFindEmployer.Form.Visible = False
Me!subFindEmployerPTS.Form.Visible = True

End Select
End Sub

Private Sub grpSearch_BeforeUpdate(Cancel As Integer)

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
```

Form: frmFindNarcs

Code

```
Attribute VB_Name = "Form_frmFindNarcs"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClear_Click()

Dim strSql As String
Dim strFullSql As String
Me!txtFind = ""
Me!cboType = ""

strSql = "Where False"
strFullSql = "Select * From qFindNarc " & strSql
Me!subFindNarc.Form.RecordSource = strFullSql

End Sub

Private Sub cmdSubmit_Click()

Dim strType As String
Dim strNarc As String
Dim strBoth As String
Dim strSql As String
Dim strMsg As String
Dim strTitle As String



Select Case grpSearch

Case 1
    strType = "Where [NarcType]= " & "Forms!frmFindNarcs![cboType]"
    strSql = "Select • From qFindNarc " & strType
    Me!subFindNarc.Form.RecordSource = strSql

Case 2
    strNarc = "Where [Brand]= " & "Forms!frmFindNarcs![txtFind]"
    strSql = "Select * From qFindNarc " & strNarc
    Me!subFindNarc.Form.RecordSource = strSql


Case 3
    strType = "[NarcType]= " & "Forms!frmFindNarcs![cboType]"
    strNarc = "[Brand]= " & "Forms!frmFindNarcs![txtFind]"
    strBoth = "Where " & strType & " AND " & strNarc
    strSql = "Select • From qFindNarc " & strBoth
    Me!subFindNarc.Form.RecordSource = strSql


End Select

End Sub

Private Sub Detail_Click()

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close
```

```

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub

Private Sub Form_Open(Cancel As Integer)


Select Case grpSearch

Case 1
    cboType.BackColor = 16777215
    txtFind.BackColor = 12632256
    Me.cboType.SetFocus
Case 2
    txtFind.BackColor = 16777215
    cboType.BackColor = 12632256
    Me.txtFind.SetFocus
Case 3
    cboType.BackColor = 16777215
    txtFind.BackColor = 16777215
    Me.cboType.SetFocus
End Select

End Sub

Private Sub grpSearch_AfterUpdate()
Select Case grpSearch

Case 1
    cboType.BackColor = 16777215
    txtFind.BackColor = 12632256
    Me.cboType.SetFocus
    Me!txtFind = ""

Case 2
    txtFind.BackColor = 16777215
    cboType.BackColor = 12632256
    Me.txtFind.SetFocus
    Me!cboType = ""

Case 3
    cboType.BackColor = 16777215
    txtFind.BackColor = 16777215
    Me.cboType.SetFocus
End Select

End Sub

Private Sub grpSearch_BeforeUpdate(Cancel As Integer)

End Sub
```

Form: frmPrisDetails

Code

```
Attribute VB_Name = "Form_frmPrisDetails"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdPreview_Click()
On Error GoTo Err_cmdPreview_Click

    Dim stDocName As String

    stDocName = "rptPrisonerDetails"
    DoCmd.OpenReport stDocName, acPreview

    DoCmd.Close acForm, "frmPrisDetails", acSaveNo

Exit_cmdPreview_Click:
    Exit Sub

Err_cmdPreview_Click:
    MsgBox Err.Description
    Resume Exit_cmdPreview_Click

End Sub

Private Sub cmdSetDate_Click()
'set date equal to date on calender
txtDate = dwnldDate.Value

End Sub

Private Sub dwnldDate_Updated(Code As Integer)

End Sub

Private Sub Form_Current()

End Sub

Private Sub Form_Load()
'set controls equal to today
Me!dwnldDate = Date
Me!txtDate = Date

End Sub
```

```
Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click

    Dim stDocName As String

    stDocName = "rptPrisonerDetails"
    DoCmd.OpenReport stDocName, acNormal

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub
Private Sub cmdCancel_Click()
On Error GoTo Err_cmdCancel_Click


    DoCmd.Close

Exit_cmdCancel_Click:
    Exit Sub

Err_cmdCancel_Click:
    MsgBox Err.Description
    Resume Exit_cmdCancel_Click

End Sub
```

Form: frmPTS

Code

```
Attribute VB_Name = "Form_frmPTS"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub

Private Sub cmdFindUSMS_Click()
Dim rst As Recordset

    Dim strCriteria As String
    Dim strUSMS As String

        'Make sure there is an entry in the text box before running
        If IsNull(txtFindUSMS) Then
            MsgBox "Enter a USMS numer first and then submit query"
            Exit Sub
        End If



        Set rst = Me.RecordsetClone
        strUSMS = Me!txtFindUSMS
        strCriteria = "[pr_usms_no]='" & strUSMS & "'"
        rst.FindFirst strCriteria
        If rst.NoMatch Then
            MsgBox "The number you entered was not found."
        Else
        Me.Bookmark = Me.RecordsetClone.Bookmark
        End If
        Me!txtFindUSMS.Value = ""

End Sub

Private Sub cmdNext_Click()
On Error GoTo Err_cmdNext_Click


    DoCmd.GoToRecord , , acNext

Exit_cmdNext_Click:
    Exit Sub

Err_cmdNext_Click:
    MsgBox Err.Description
    Resume Exit_cmdNext_Click

End Sub
Private Sub cmdPrevious_Click()
```

```
On Error GoTo Err_cmdPrevious_Click


    DoCmd.GoToRecord , , acPrevious

Exit_cmdPrevious_Click:
    Exit Sub

Err_cmdPrevious_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrevious_Click

End Sub
Sub Combo33_AfterUpdate()

    ' Find the record that matches the control.
    If IsNull(Me![Combo33]) Then
        Exit Sub
    End If

    ' Find the record that matches the control.
    Me.RecordsetClone.FindFirst "[pr_key] = " & Me![Combo33]
    Me.Bookmark = Me.RecordsetClone.Bookmark
End Sub

Private Sub Form_Current()

'synchronize combo box with record
Combo33 = pr_key
End Sub
Private Sub cmdPrintReport_Click()
On Error GoTo Err_cmdPrintReport_Click

    Dim stDocName As String

    stDocName = "rptPrisonerDetailsPTS"
    DoCmd.OpenReport stDocName, acNormal

Exit_cmdPrintReport_Click:
    Exit Sub

Err_cmdPrintReport_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrintReport_Click

End Sub
```

Code

```
Attribute VB_Name = "Form_frmPTSArrests"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit
Const ConErrRptCanceled = 2501


Private Sub cmdCancel_Click()

' Close form
On Error GoTo Err_cmdCancel_Click

    DoCmd.Close

Exit_cmdCancel_Click:
    Exit Sub

Err_cmdCancel_Click:
    MsgBox Err.Description
    Resume Exit_cmdCancel_Click

End Sub



Private Sub cmdPreview_Click()

' Preview report.
Dim strDocName As String

    On Error GoTo Err_cmdPreview_Click

    ' Check to see that ending date is later than beginning date.
    If IsDate(BeginDate) And IsDate(EndDate) Then
        If EndDate < BeginDate Then
            MsgBox "The end date must be later than the begin date."
            cmdSetDate.Caption = "Set End Date"
            SelectDate.SetFocus
            Exit Sub
        End If
    Else
        MsgBox "Please use a valid date for the beginning date and the ending date values."

Exit Sub

    End If

    strDocName = "PTS Arrests"
    DoCmd.OpenReport strDocName, acViewPreview

Exit_cmdPreview_Click:
    Exit Sub

Err_cmdPreview_Click:
    If Err = ConErrRptCanceled Then
        Resume Exit_cmdPreview_Click
    Else
        MsgBox Err.Description
        Resume Exit_cmdPreview_Click
    End If

End Sub
```

Form: frmPTSArrests

```

Private Sub cmdPrint_Click()
    ' Print report.
    Dim strDocName As String

    On Error GoTo Err_cmdPrint_Click

    ' Check to see that ending date is later than beginning date.
    If EndDate < BeginDate Then
        MsgBox "The end date must be later than the begin date."
        cmdSetDate.Caption = "Set End Date"
        SelectDate.SetFocus
        Exit Sub
    End If

    strDocName = "PTS Arrests"
    DoCmd.OpenReport strDocName, acViewNormal

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    If Err = ConErrRptCanceled Then
        Resume Exit_cmdPrint_Click
    Else
        MsgBox Err.Description
        Resume Exit_cmdPrint_Click
    End If

End Sub


Private Sub cmdSetDate_Click()

    On Error GoTo cmdSetDate_Error

    If cmdSetDate.Caption = "Set Begin Date" Then
        BeginDate = SelectDate.Value
        cmdSetDate.Caption = "Set End Date"
    Else
        EndDate = SelectDate.Value
        cmdSetDate.Caption = "Set Begin Date"
    End If

    Exit Sub

cmdSetDate_Error:
    MsgBox Err.Description
    Exit Sub

End Sub

Function IsLoaded(strfrmName As String) As Integer

    ' Determines if a form is loaded.

    Const conFormDesign = 0
    Dim intX As Integer

    IsLoaded = False
    For intX = 0 To Forms.Count - 1
        If Forms(intX).FormName = strfrmName Then
            If Forms(intX).CurrentView <> conFormDesign Then
                IsLoaded = True
                Exit Function       ' Quit function once form has been found.
            End If
        End If
```

This document is a research report submitted to the U.S. Department of Justice. This report has not been published by the Department. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.

```
    'Next

```

Form: frmPTSArrests

```
End Function


Private Sub Detail_Click()

End Sub

Private Sub Form_Load()

'Enter parameters in date boxes
Me!SelectDate = Now()
Me!BeginDate = Now() - 7
Me!EndDate = Now()

End Sub
```

Form: frmREPORTS

Code

```
Attribute VB_Name = "Form_frmREPORTS"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdOpenRptPtsDwn_Click()
On Error GoTo Err_cmdOpenRptPtsDwn_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "PTSDownloads"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenRptPtsDwn_Click:
    Exit Sub

Err_cmdOpenRptPtsDwn_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenRptPtsDwn_Click

End Sub
```

Form: frmTARGET_PHOTOsub

Code

```
Attribute VB_Name = "Form_frmTARGET_PHOTOsub"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click


    DoCmd.PrintOut

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub
```

Form: frmTARGETS

Code

```
Attribute VB_Name = "Form_frmTARGETS"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdPhoto_Click()
On Error GoTo Err_cmdPhoto_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmTARGET_PHOTOsub"

    stLinkCriteria = "[TargetKey]=" & Me![TargetKey]

    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdPhoto_Click:
    Exit Sub

Err_cmdPhoto_Click:
    MsgBox Err.Description
    Resume Exit_cmdPhoto_Click

End Sub

Private Sub Form_AfterUpdate()

'enable or disable photo buttons based on record status
    If IsNull(Me![TargetKey]) And IsNull(Me![txtPhoto]) Then
        Me!cmdPhoto.Enabled = False
    Else
        Me!cmdPhoto.Enabled = True
    End If

    If Not (IsNull(Me![TargetKey])) And Not (IsNull(Me![txtPhoto])) Then
        Me!cmdInsertPhoto.Enabled = True
    Else
        Me!cmdInsertPhoto.Enabled = False
    End If
'enable or disable photo buttons based on record status
    If IsNull(Me![TargetKey]) And IsNull(Me![txtPhoto]) Then
        Me!cmdPhoto.Enabled = False
    Else
        Me!cmdPhoto.Enabled = True
    End If

    If Not (IsNull(Me![TargetKey])) And Not (IsNull(Me![txtPhoto])) Then
        Me!cmdInsertPhoto.Enabled = True
    Else
        Me!cmdInsertPhoto.Enabled = False
    End If

End Sub

Private Sub Form_Current()

'enable or disable photo buttons based on record status
    If IsNull(Me![TargetKey]) Or IsNull(Me![txtPhoto]) Then
        Me!cmdPhoto.Enabled = False
    Else
        Me!cmdPhoto.Enabled = True
    End If

    If Not (IsNull(Me![TargetKey])) And (IsNull(Me![txtPhoto])) Then
        Me!cmdInsertPhoto.Enabled = True
    Else
        Me!cmdInsertPhoto.Enabled = False
    End If

End Sub

Private Sub Form_Load()
    If IsNull(Me![TargetKey]) Then
        Me!cmdPhoto.Enabled = False
    Else
        Me!cmdPhoto.Enabled = True
    End If

End Sub
Private Sub cmdInsertPhoto_Click()
On Error GoTo Err_cmdInsertPhoto_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmTARGET_PHOTOsub"

    stLinkCriteria = "[TargetKey]=" & Me![TargetKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdInsertPhoto_Click:
    Exit Sub

Err_cmdInsertPhoto_Click:
    MsgBox Err.Description
    Resume Exit_cmdInsertPhoto_Click

End Sub
```

Form: HOMICIDEsubform

Code

```
Attribute VB_Name = "Form_HOMICIDEsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Form_Open(Cancel As Integer)

'On opening set focus to Homicide date
Me!HomDate.SetFocus

End Sub
```

Form: LOCALEsubform

Code

```
Attribute VB_Name = "Form_LOCALEsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Form_Load()

Me!LocName.SetFocus
Me.Repaint

End Sub
```

Form: NARCOTICSsubform

Code

```
Attribute VB_Name = "Form_NARCOTICSsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Form_Current()

End Sub

Private Sub NarcType_AfterUpdate()

If IsNull(Me!NarcType) Then
    Forms!COOPERATORS!subNARCOTICS!subNARC_BRAND!Brand.Enabled = False
Else
    Forms!COOPERATORS!subNARCOTICS!subNARC_BRAND!Brand.Enabled = True
End If

End Sub

Private Sub NarcType_BeforeUpdate(Cancel As Integer)

End Sub
```

Code

```
Attribute VB_Name = "Form_PTSDownloads"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit
Const ConErrRptCanceled = 2501


Private Sub cmdCancel_Click()

' Close form
On Error GoTo Err_cmdCancel_Click

    DoCmd.Close

Exit_cmdCancel_Click:
    Exit Sub

Err_cmdCancel_Click:
    MsgBox Err.Description
    Resume Exit_cmdCancel_Click

End Sub



Private Sub cmdPreview_Click()

' Preview report.
Dim strDocName As String

    On Error GoTo Err_cmdPreview_Click

    ' Check to see that ending date is later than beginning date.
    If IsDate(BeginDate) And IsDate(EndDate) Then
        If EndDate < BeginDate Then
            MsgBox "The end date must be later than the begin date."
            cmdSetDate.Caption = "Set End Date"
            SelectDate.SetFocus
            Exit Sub
        End If
    Else
        MsgBox "Please use a valid date for the beginning date and the ending date values."

Exit Sub
    End If

    strDocName = "PTS Downloads"
    DoCmd.OpenReport strDocName, acViewPreview

Exit_cmdPreview_Click:
    Exit Sub

Err_cmdPreview_Click:
    If Err = ConErrRptCanceled Then
        Resume Exit_cmdPreview_Click
    Else
        MsgBox Err.Description
        Resume Exit_cmdPreview_Click
    End If

End Sub
```

Form: PTSDownloads

```

Private Sub cmdPrint_Click()
    ' Print report.
    Dim strDocName As String

    On Error GoTo Err_cmdPrint_Click

    ' Check to see that ending date is later than beginning date.
    If EndDate < BeginDate Then
        MsgBox "The end date must be later than the begin date."
        cmdSetDate.Caption = "Set End Date"
        SelectDate.SetFocus
        Exit Sub
    End If

    strDocName = "PTS Downloads"
    DoCmd.OpenReport strDocName, acViewNormal

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    If Err = ConErrRptCanceled Then
        Resume Exit_cmdPrint_Click
    Else
        MsgBox Err.Description
        Resume Exit_cmdPrint_Click
    End If

End Sub


Private Sub cmdSetDate_Click()

    On Error GoTo cmdSetDate_Error

    If cmdSetDate.Caption = "Set Begin Date" Then
        BeginDate = SelectDate.Value
        cmdSetDate.Caption = "Set End Date"
    Else
        EndDate = SelectDate.Value
        cmdSetDate.Caption = "Set Begin Date"
    End If

    Exit Sub

cmdSetDate_Error:
    MsgBox Err.Description
    Exit Sub

End Sub

Function IsLoaded(strfrmName As String) As Integer

    ' Determines if a form is loaded.

    Const conFormDesign = 0
    Dim intX As Integer

    IsLoaded = False
    For intX = 0 To Forms.Count - 1
        If Forms(intX).FormName = strfrmName Then
            If Forms(intX).CurrentView <> conFormDesign Then
                IsLoaded = True
                Exit Function       ' Quit function once form has been found.
            End If
        End If
    'Next

```

```
End Function


Private Sub Detail_Click()

End Sub

Private Sub Form_Load()

'Enter parameters in date boxes
Me!SelectDate = Now()
Me!BeginDate = Now() - 7
Me!EndDate = Now()

End Sub
```

Form: StartUp

Code

```
Attribute VB_Name = "Form_StartUp"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdOpenAlias_Click()
On Error GoTo Err_cmdOpenAlias_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmFindAlias"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenAlias_Click:
    Exit Sub

Err_cmdOpenAlias_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenAlias_Click

End Sub

Private Sub cmdOpenAssoc_Click()
On Error GoTo Err_cmdOpenAssoc_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmFindAssoc"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenAssoc_Click:
    Exit Sub

Err_cmdOpenAssoc_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenAssoc_Click

End Sub

Private Sub cmdOpenCoopFrm_Click()
On Error GoTo Err_cmdOpenCoopFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "COOPERATORS"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenCoopFrm_Click:
    Exit Sub

Err_cmdOpenCoopFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCoopFrm_Click


End Sub
```

```

Private Sub cmdOpenCounsel_Click()
On Error GoTo Err_cmdOpenCounsel_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmFindCounsel"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenCounsel_Click:
    Exit Sub

Err_cmdOpenCounsel_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCounsel_Click

End Sub

Private Sub cmdOpenDefFrm_Click()
On Error GoTo Err_cmdOpenDefFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "DEFENDANT"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenDefFrm_Click:
    Exit Sub

Err_cmdOpenDefFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenDefFrm_Click


End Sub

Private Sub cmdOpenFindEmploy_Click()

End Sub

Private Sub cmdOpenMaint_Click()

Me.GoToPage 4

End Sub

Private Sub cmdOpenNarc_Click()
On Error GoTo Err_cmdOpenNarc_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmFindNarcs"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenNarc_Click:
    Exit Sub

Err_cmdOpenNarc_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenNarc_Click

End Sub
```

```
Private Sub cmdOpenPg3_Click()

Me.GoToPage 3

End Sub

Private Sub cmdOpenRptfrm_Click()
Me.GoToPage 2
End Sub

Private Sub cmdOpenTQfrm_Click()

If Me!lblMain.Visible = True Then
    Me.lblMain.Visible = False
    Me.lblRpt.Visible = False
    Me.lblSQ.Visible = False
    Me.lblTQ.Visible = False
    Me.lblRecords.Visible = False
    Me.lblMaint.Visible = False
    Me.cmdOpenMaint.Visible = False
    Me.cmdOpenPg3.Visible = False
    Me.cmdOpenSQfrm.Visible = False
    Me.cmdOpenRptfrm.Visible = False
    Me.lblAlias.Visible = True
    Me.lblNarc.Visible = True
    Me.lblAssoc.Visible = True
    Me.lblCounsel.Visible = True
    Me.lblEmploy.Visible = True
    Me.lblReturnMain.Visible = True
    Me.cmdOpenAlias.Visible = True
    Me.cmdOpenNarc.Visible = True
    Me.cmdOpenAssoc.Visible = True
    Me.cmdOpenCounsel.Visible = True
    Me.cmdOpenEmploy.Visible = True
    Me.cmdOpenAlias.SetFocus
    Me.cmdOpenTQfrm.Picture = "c:\Program
Files\DevStudio\VB\Graphics\Icons\Arrows\Arw09lt.ico"

ElseIf Me!lblMain.Visible = False Then
    Me.lblMain.Visible = True
    Me.lblRpt.Visible = True
    Me.lblSQ.Visible = True
    Me.lblTQ.Visible = True
    Me.lblRecords.Visible = True
    Me.cmdOpenPg3.Visible = True
    Me.cmdOpenSQfrm.Visible = True
    Me.cmdOpenRptfrm.Visible = True
    Me.lblMaint.Visible = True
    Me.cmdOpenMaint.Visible = True
    Me.lblAlias.Visible = False
    Me.lblNarc.Visible = False
    Me.lblAssoc.Visible = False
    Me.lblCounsel.Visible = False
    Me.lblEmploy.Visible = False
    Me.lblReturnMain.Visible = False
    Me.cmdOpenAlias.Visible = False
    Me.cmdOpenNarc.Visible = False
    Me.cmdOpenAssoc.Visible = False
    Me.cmdOpenCounsel.Visible = False
    Me.cmdOpenEmploy.Visible = False
    Me.cmdOpenTQfrm.Picture = "C:\Program
Files\DevStudio\VB\Graphics\Icons\Misc\LIGHTON.ICO"
End If
End Sub

Private Sub Command30_Click()
```

```
Me.GoToPage 1
End Sub
```

Form: StartUp

```

Private Sub Command34_Click()

End Sub

Private Sub Command40_Click()

Me.GoToPage 1

End Sub

Private Sub Command82_Click()

'run transfer.exe - vb app that inputs email messages into database
Dim RetVal
RetVal = Shell("D:\NETWORK\RACKETS\MISC\TRANSLATE.EXE", 1)

DoCmd.Close acForm, "StartUp"

End Sub

Private Sub Form_Load()
Me.GoToPage 1

Me.lblAlias.Visible = False
Me.lblNarc.Visible = False
Me.lblAssoc.Visible = False
Me.lblCounsel.Visible = False
Me.lblEmploy.Visible = False
Me.lblReturnMain.Visible = False
Me.cmdOpenAlias.Visible = False
Me.cmdOpenNarc.Visible = False
Me.cmdOpenAssoc.Visible = False
Me.cmdOpenCounsel.Visible = False
Me.cmdOpenEmploy.Visible = False
Me.lblMain.Visible = True
Me.lblRpt.Visible = True
Me.lblSQ.Visible = True
Me.lblTQ.Visible = True
Me.lblRecords.Visible = True
Me.lblMaint.Visible = True
Me.cmdOpenMaint.Visible = True
Me.cmdOpenPg3.Visible = True
Me.cmdOpenTQfrm.Visible = True
Me.cmdOpenSQfrm.Visible = True
Me.cmdOpenRptfrm.Visible = True

End Sub

Private Sub Form_Open(Cancel As Integer)

Me.GoToPage 1

Me.lblAlias.Visible = False
Me.lblNarc.Visible = False
Me.lblAssoc.Visible = False
Me.lblCounsel.Visible = False
Me.lblEmploy.Visible = False
Me.lblReturnMain.Visible = False
Me.cmdOpenAlias.Visible = False
Me.cmdOpenNarc.Visible = False
Me.cmdOpenAssoc.Visible = False
Me.cmdOpenCounsel.Visible = False
Me.cmdOpenEmploy.Visible = False
Me.lblMain.Visible = True
Me.lblRpt.Visible = True
Me.lblSQ.Visible = True
Me.lblTQ.Visible = True
Me.lblMaint.Visible = True
```

Form: StartUp

```
Me.cmdOpenMaint.Visible = True
Me.lblRecords.Visible = True
Me.cmdOpenPg3.Visible = True
Me.cmdOpenTQfrm.Visible = True
Me.cmdOpenSQfrm.Visible = True
Me.cmdOpenRptfrm.Visible = True

End Sub
Private Sub cmdExit_Click()
On Error GoTo Err_cmdExit_Click


    DoCmd.Quit

Exit_cmdExit_Click:
    Exit Sub

Err_cmdExit_Click:
    MsgBox Err.Description
    Resume Exit_cmdExit_Click

End Sub
Private Sub cmdOpenCaseFrm_Click()
On Error GoTo Err_cmdOpenCaseFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "CASE"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenCaseFrm_Click:
    Exit Sub

Err_cmdOpenCaseFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCaseFrm_Click

End Sub
Private Sub cmdOpenNewCase_Click()
On Error GoTo Err_cmdOpenNewCase_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "CASE"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

    Forms!CASE.DataEntry = True
    Forms!CASE.cmdArchive.Enabled = False
    Forms!CASE.cmdArchive.Visible = False
    Forms!CASE.Combo15.Enabled = False
    Forms!CASE.Combo15.Visible = False
    Forms!CASE.lblUSV.Visible = False

Exit_cmdOpenNewCase_Click:
    Exit Sub

Err_cmdOpenNewCase_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenNewCase_Click

End Sub
```

```
Private Sub cmdOpenPTS_Click()
On Error GoTo Err_cmdOpenPTS_Click

    Dim stDocName As String
```

```
    Dim stLinkCriteria As String

    stDocName = "frmPTS"
    DoCmd.Close acForm, "StartUp"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenPTS_Click:
    Exit Sub

Err_cmdOpenPTS_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenPTS_Click

End Sub
Private Sub cmdOpenEmploy_Click()
On Error GoTo Err_cmdOpenEmploy_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmFindEmployer"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenEmploy_Click:
    Exit Sub

Err_cmdOpenEmploy_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenEmploy_Click

End Sub
Private Sub cmdOpenclnArrest_Click()
On Error GoTo Err_cmdOpenclnArrest_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "cln_ARREST"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenclnArrest_Click:
    Exit Sub

Err_cmdOpenclnArrest_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenclnArrest_Click

End Sub
Private Sub cmdOpenPTSArr_Click()
End Sub
```

Form: STATUTESsubform

Code

```
Attribute VB_Name = "Form_STATUTESsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Count_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub Form_AfterUpdate()
End Sub

Private Sub Form_BeforeUpdate(Cancel As Integer)


End Sub

Private Sub Form_Current()


End Sub

Private Sub Form_Error(DataErr As Integer, Response As Integer)

    Const conErrFieldRequired = 3314

    If DataErr = conErrFieldRequired Then
        DisplayMessage "You must enter a Count and Statute number to save the record."
        Response = acDataErrContinue
    Else
        Response = acDataErrDisplay
    End If

End Sub

Private Sub Form_Load()

End Sub

Private Sub Location_AfterUpdate()


End Sub

Private Sub Location_Click()


End Sub

Private Sub StatuteDesc_AfterUpdate()



End Sub
```

Form: WARRANTsubform

Code

```
Attribute VB_Name = "Form_WARRANTsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Form_Open(Cancel As Integer)


End Sub
```

Form: WIRETAPsubform

Code

```
Attribute VB_Name = "Form_WIRETAPsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Form_Open(Cancel As Integer)



End Sub
```

Form: CASE

Code

```
Attribute VB_Name = "Form_CASE"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit


Private Sub AUSA_NotInList(NewData As String, Response As Integer)

Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of AUSAs?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("ausa")
    rstTypes.AddNew
    rstTypes!AUSA = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If

End Sub

Sub Combo15_AfterUpdate()
    ' Find the record that matches the control.
    If IsNull(Me![Combo15]) Then
        Exit Sub
    End If

    Me.RecordsetClone.FindFirst "[CaseKey] = " & Me![Combo15]
    Me.Bookmark = Me.RecordsetClone.Bookmark
End Sub



Private Sub Command50_Click()

End Sub

Private Sub Combo15_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub Form_Current()
    Combo15 = CaseKey
End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
Private Sub cmdPrevious_Click()
On Error GoTo Err_cmdPrevious_Click


    DoCmd.GoToRecord , , acPrevious

Exit_cmdPrevious_Click:
    Exit Sub

Err_cmdPrevious_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrevious_Click

End Sub
Private Sub cmdNext_Click()
On Error GoTo Err_cmdNext_Click


    DoCmd.GoToRecord , , acNext

Exit_cmdNext_Click:
    Exit Sub

Err_cmdNext_Click:
    MsgBox Err.Description
    Resume Exit_cmdNext_Click

End Sub
Private Sub cmdAddRecord_Click()
```

```
On Error GoTo Err_cmdAddRecord_Click


    DoCmd.GoToRecord , , acNewRec
    Me!USAONum.SetFocus

Exit_cmdAddRecord_Click:
    Exit Sub

Err_cmdAddRecord_Click:
    MsgBox Err.Description
    Resume Exit_cmdAddRecord_Click

End Sub
Private Sub cmdArchive_Click()
On Error GoTo Err_cmdArchive_Click

    Dim stDocName As String
    Dim stDocName1 As String
    Dim stDocName2 As String
    Dim stDocName3 As String
    Dim stDocName4 As String
    Dim stDocName5 As String
    Dim stDocName6 As String

    stDocName = "arcCASE"
    stDocName1 = "arcDEFENDANT"
    stDocName2 = "arcALIAS"
    stDocName3 = "arcSTATUTES"
    stDocName4 = "arcEVENTS"
    stDocName5 = "arcGROUPS"
    stDocName6 = "delCASE"

    DoCmd.OpenQuery stDocName, acNormal, acEdit
    DoCmd.OpenQuery stDocName1, acNormal, acEdit
    DoCmd.OpenQuery stDocName2, acNormal, acEdit
    DoCmd.OpenQuery stDocName3, acNormal, acEdit
    DoCmd.OpenQuery stDocName4, acNormal, acEdit
    DoCmd.OpenQuery stDocName5, acNormal, acEdit
    DoCmd.OpenQuery stDocName6, acNormal, acEdit

Exit_cmdArchive_Click:
    Exit Sub

Err_cmdArchive_Click:
    MsgBox Err.Description
    Resume Exit_cmdArchive_Click

End Sub

Private Sub Form_Error(DataErr As Integer, Response As Integer)
```

```

Const conErrFieldRequired = 3314

If DataErr = conErrFieldRequired Then
    DisplayMessage "You must enter a Case name and USAO number to save the record."
    Response = acDataErrContinue
Else
    Response = acDataErrDisplay
End If

End Sub

Private Sub Form_Load()
End Sub
```

Form: COOP_AREA

Code

```
Attribute VB_Name = "Form_COOP_AREA"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Topic_NotInList(NewData As String, Response As Integer)
Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "' to the list of Activities?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("coop_activity")
    rstTypes.AddNew
    rstTypes!Activity = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If

End Sub
```

Form: COOP_PEOPLEsubArea

Code

```
Attribute VB_Name = "Form_COOP_PEOPLEsubArea"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit



Private Sub Form_BeforeUpdate(Cancel As Integer)
Me![CoopKey] = Me.Parent![CoopKey]
End Sub
```

Form: COOPERATORS

Code

```
Attribute VB_Name = "Form_COOPERATORS"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cboFindName_AfterUpdate()

Me.RecordsetClone.FindFirst "[DefKey] = " & Me![cboFindName]
Me.Bookmark = Me.RecordsetClone.Bookmark


End Sub

Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub

Private Sub LName_AfterUpdate()

'Fill in Last name from defendant table
strFilter = "Me!DefKey = Forms!DEFENDANT!DefKey"
Me!LName = DLookup("[DefLName]", "DEFENDANT", strFilter)

End Sub

Private Sub LName_BeforeUpdate(Cancel As Integer)

End Sub
Private Sub cmdArchive_Click()
On Error GoTo Err_cmdArchive_Click

    Dim stDocName As String
    Dim stDocName1 As String
    Dim stDocName2 As String
    Dim stDocName3 As String
    Dim stDocName4 As String
    Dim stDocName5 As String

    stDocName = "arcCOOPERATOR"
    stDocName1 = "arcCOOP_AREA"
    stDocName2 = "arcCOOP_GROUP"
    stDocName3 = "arcHOMICIDE"
    stDocName4 = "arcNARCOTIC"
    stDocName5 = "delCOOPERATOR"

    DoCmd.OpenQuery stDocName, acNormal, acEdit
    DoCmd.OpenQuery stDocName1, acNormal, acEdit
    DoCmd.OpenQuery stDocName2, acNormal, acEdit
    DoCmd.OpenQuery stDocName3, acNormal, acEdit
    DoCmd.OpenQuery stDocName4, acNormal, acEdit
    DoCmd.OpenQuery stDocName5, acNormal, acEdit
```

```

Exit_cmdArchive_Click:
    Exit Sub

Err_cmdArchive_Click:
    MsgBox Err.Description
    Resume Exit_cmdArchive_Click

End Sub

Private Sub Combo42_AfterUpdate()

End Sub

Private Sub Form_Current()

'Synchronize combobox and records
cboFindName = DefKey

End Sub
Private Sub cmdPrevious_Click()
On Error GoTo Err_cmdPrevious_Click


    DoCmd.GoToRecord , , acPrevious

Exit_cmdPrevious_Click:
    Exit Sub

Err_cmdPrevious_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrevious_Click

End Sub
Private Sub cmdNext_Click()
On Error GoTo Err_cmdNext_Click


    DoCmd.GoToRecord , , acNext

Exit_cmdNext_Click:
    Exit Sub

Err_cmdNext_Click:
    MsgBox Err.Description
    Resume Exit_cmdNext_Click

End Sub
Private Sub cmdFirst_Click()
On Error GoTo Err_cmdFirst_Click


    DoCmd.GoToRecord , , acFirst

Exit_cmdFirst_Click:
    Exit Sub

Err_cmdFirst_Click:
    MsgBox Err.Description
    Resume Exit_cmdFirst_Click

End Sub
Private Sub cmdLast_Click()
On Error GoTo Err_cmdLast_Click


    DoCmd.GoToRecord , , acLast

```

```
Exit_cmdLast_Click:
```

```
    Exit Sub

Err_cmdLast_Click:
    MsgBox Err.Description
    Resume Exit_cmdLast_Click

End Sub
Private Sub cmdPrntReport_Click()
On Error GoTo Err_cmdPrntReport_Click

    Dim stDocName As String

    stDocName = "COOPERATORS"
    DoCmd.OpenReport stDocName, acNormal

Exit_cmdPrntReport_Click:
    Exit Sub

Err_cmdPrntReport_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrntReport_Click

End Sub
```

Form: COOPERATORSsub

Code

```
Attribute VB_Name = "Form_COOPERATORSsub"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Topic_NotInList(NewData As String, Response As Integer)

'If item is not in list, check then add to list if requested
Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of Main Topics?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("coopTopic")
    rstTypes.AddNew
    rstTypes!Topic = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If

End Sub
```

Form: COOPERATORsubform

Code

```
Attribute VB_Name = "Form_COOPERATORsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdOpenCoopFrm_Click()
On Error GoTo Err_cmdOpenCoopFrm_Click

    'Save record first
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70

    Dim stDocName As String
    Dim stLinkCriteria As String
    'Dim stMsg As String
    'Dim stTitle As String

    'Check to see if there is a valid record selected before opening form
    'If IsNull(Me![CoopKey]) Then 'Notify user and then exit sub
    '    stMsg = "You must enter basic Cooperator information before you can select this option"
    '    stTitle = "Select a Record and then Continue"
    '        If MsgBox(stMsg, vbOKOnly, stTitle) = vbOK Then
    '            Exit Sub
    '        End If
    'End If

    stDocName = "COOPERATORS"

    stLinkCriteria = "[CoopKey]=" & Me![CoopKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenCoopFrm_Click:
    Exit Sub

Err_cmdOpenCoopFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCoopFrm_Click

End Sub

Private Sub Form_Current()

Me!CoopDate.SetFocus

End Sub

Private Sub Form_Unload(Cancel As Integer)

Me!CoopDate.SetFocus
Me.Repaint

End Sub
```

Form: CurrentPrisoners

Code

```
Attribute VB_Name = "Form_CurrentPrisoners"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Combo21_BeforeUpdate(Cancel As Integer)

End Sub


Private Sub Form_Current()

'Synchronize combobox with form
Combo21 = Me!FullName

End Sub


Private Sub Form_Load()

End Sub

Private Sub cmdImportRackets_Click()
On Error GoTo Err_cmdImportRackets_Click

'Confirm transfer
Dim strMsg As String
strMsg = "You are about to import a record from PTS to RACKETS, Continue?"
If MsgBox(strMsg, vbOKCancel, "Error!") = vbCancel Then
    Exit Sub
Else
End If

End Sub

Private Sub Form_Open(Cancel As Integer)

Me.pr_street.SetFocus

End Sub

Private Sub FormFooter_Click()

End Sub

Private Sub FormHeader_Click()

End Sub
Sub Combo21_AfterUpdate()
    ' Find the record that matches the control.
    Me.RecordsetClone.FindFirst "[FullName] = '" & Me![Combo21] & "'"
    Me.Bookmark = Me.RecordsetClone.Bookmark
End Sub


Private Sub pr_citizen_code_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub pr_citizen_code_Click()

End Sub
```

```
Private Sub cmdUpdateRec_Click()
On Error GoTo Err_cmdUpdateRec_Click

    Dim stDocName As String
    'Dim stDocName2 As String

    'Set dbs = CurrentDb
    'Set rst = dbs.OpenRecordset("DEFENDANT", dbOpenDynaset)

    stDocName = "qUpdateWPTS"
    'stDocName2 = "qApndArrest"

    DoCmd.OpenQuery stDocName, acNormal, acEdit

Exit_cmdUpdateRec_Click:
    Exit Sub

Err_cmdUpdateRec_Click:
    MsgBox Err.Description
    Resume Exit_cmdUpdateRec_Click

End Sub
Private Sub cmdNext_Click()
On Error GoTo Err_cmdNext_Click


    DoCmd.GoToRecord , , acNext

Exit_cmdNext_Click:
    Exit Sub

Err_cmdNext_Click:
    MsgBox Err.Description
    Resume Exit_cmdNext_Click

End Sub
Private Sub cmdPrevious_Click()
On Error GoTo Err_cmdPrevious_Click


    DoCmd.GoToRecord , , acPrevious

Exit_cmdPrevious_Click:
    Exit Sub

Err_cmdPrevious_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrevious_Click

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
Private Sub cmdArrest_Click()
```

```
On Error GoTo Err_cmdArrest_Click

    Dim stDocName As String

    stDocName = "qApndArrest"
    DoCmd.OpenQuery stDocName, acNormal, acEdit

Exit_cmdArrest_Click:
    Exit Sub

Err_cmdArrest_Click:
    MsgBox Err.Description
    Resume Exit_cmdArrest_Click

End Sub
```

Form: DEFENDANT

Code

```
Attribute VB_Name = "Form_DEFENDANT"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cboFindName_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub

Private Sub cmdSubmit_Click()

Dim rst As Recordset

    Dim strCriteria As String
    Dim strNYSID As String
    Dim strUSMS As String

        'Make sure there is an entry in the text box before running
        If IsNull(txtFindBy) Then
            MsgBox "Enter a NYSID or USMS numer and then submit query"
            Exit Sub
        End If

Select Case grpFindOptions


        Case 1 'Find case using NYSID number
            Set rst = Me.RecordsetClone
            strNYSID = Me!txtFindBy
            strCriteria = "[NYSID]='" & strNYSID & "'"
            rst.FindFirst strCriteria
            If rst.NoMatch Then
                MsgBox "The number you entered is not valid."
            Else
            Me.Bookmark = Me.RecordsetClone.Bookmark
            End If
            Me!txtFindBy.Value = ""
        Case 2 'Find case using USMS number
            Set rst = Me.RecordsetClone
            strUSMS = Me!txtFindBy
            strCriteria = "[USMSNum]='" & strUSMS & "'"
            rst.FindFirst strCriteria
            If rst.NoMatch Then
                MsgBox "The number you entered is not valid."
            Else
            Me.Bookmark = Me.RecordsetClone.Bookmark
            End If
            Me!txtFindBy.Value = ""
End Select

End Sub

Sub cboFindName_AfterUpdate()

    Dim rst As Recordset

    ' Find the record that matches the control.
    If IsNull(Me![cboFindName]) Then
        Exit Sub
    End If

    Set rst = Me.RecordsetClone
    rst.FindFirst "[DefKey] = " & Me![cboFindName]
    'Me.RecordsetClone.FindFirst "[DefKey] = " & Me![cboFindName]
    If rst.NoMatch Then
        rst.Close
        Exit Sub
    End If
    Me.Bookmark = rst.Bookmark

End Sub

Private Sub cmdNext_Click()
On Error GoTo Err_cmdNext_Click


    DoCmd.GoToRecord , , acNext

Exit_cmdNext_Click:
    Exit Sub

Err_cmdNext_Click:
    MsgBox Err.Description
    Resume Exit_cmdNext_Click

End Sub
Private Sub cmdPrevious_Click()
On Error GoTo Err_cmdPrevious_Click


    DoCmd.GoToRecord , , acPrevious

Exit_cmdPrevious_Click:
    Exit Sub

Err_cmdPrevious_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrevious_Click

End Sub

Private Sub Coop_AfterUpdate()

'If cooperator is true set cooperator page visible
If (Me!Coop = True) Then
    Me!tpgCooperator.Visible = True

ElseIf (Me!Coop = False) Then
    Me!tpgCooperator.Visible = False
Else
End If

End Sub

Private Sub Coop_BeforeUpdate(Cancel As Integer)
```

```


End Sub

Private Sub Coop_Click()

End Sub

Private Sub CrimeType_NotInList(NewData As String, Response As Integer)

Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of Crime Types?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("crime_type")
    rstTypes.AddNew
    rstTypes!CrimeType = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If

End Sub

Private Sub Detail_Click()

End Sub

Private Sub Form_AfterUpdate()

'Enable or disable mugshots button
If IsNull(Me!pr_key) Or IsNull(Me!txtMugshot) Then
    Me!cmdPhoto.Enabled = False
Else
    Me!cmdPhoto.Enabled = True
End If

'Enable or disable insert mugshots button
If IsNull(Me!txtMugshot) And Not (IsNull(Me!pr_key)) Then
    Me!cmdEnterPhoto.Enabled = True
Else
    Me!cmdEnterPhoto.Enabled = False
End If

End Sub

Private Sub Form_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub Form_Current()

'Set Find Record to current
cboFindName = DefKey

'If cooperator is true set cooperator page visible
If (Me!Coop = True) Then
    Me!tpgCooperator.Visible = True

ElseIf (Me!Coop = False) Then
```

```
    Me!tpgCooperator.Visible = False
Else
End If

'If Weapon seized is true set cooperator page visible
If (Me!Weapon = True) Then
    Me!tpgFIREARMS.Visible = True

ElseIf (Me!Weapon = False) Then
    Me!tpgFIREARMS.Visible = False
Else
End If

'Enable or disable mugshots button
If IsNull(Me!pr_key) Or IsNull(Me!txtMugshot) Then
    Me!cmdPhoto.Enabled = False
Else
    Me!cmdPhoto.Enabled = True
End If

'Enable or disable insert mugshots button
If IsNull(Me!txtMugshot) And Not (IsNull(Me!pr_key)) Then
    Me!cmdEnterPhoto.Enabled = True
Else
    Me!cmdEnterPhoto.Enabled = False
End If

End Sub
Private Sub cmdOpenCooperatorFrm_Click()
On Error GoTo Err_cmdOpenCooperatorFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "COOPERATORS"

    stLinkCriteria = "[DefKey]=" & Me![DefKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenCooperatorFrm_Click:
    Exit Sub

Err_cmdOpenCooperatorFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCooperatorFrm_Click

End Sub
Private Sub cmdOpenCooperatorForm_Click()
On Error GoTo Err_cmdOpenCooperatorForm_Click

    Dim stDocName As String

    Dim stLinkCriteria As String

    stDocName = "COOPERATORS"

    stLinkCriteria = "[DefKey]=" & Me!DefKey
    DoCmd.OpenForm stDocName, , , stLinkCriteria, acAdd, , Me!DefKey


Exit_cmdOpenCooperatorForm_Click:
    Exit Sub

Err_cmdOpenCooperatorForm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCooperatorForm_Click

```

```
End Sub
```

Form: DEFENDANT

```

Private Sub Form_Error(DataErr As Integer, Response As Integer)
    Debug.Print "DataErr = "; DataErr
End Sub


Private Sub Frame99_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub Frame112_BeforeUpdate(Cancel As Integer)

End Sub


Private Sub grpFindOptions_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub LeadDef_AfterUpdate()

End Sub

Private Sub LeadDef_Click()

End Sub

Private Sub txtFindBy_BeforeUpdate(Cancel As Integer)

End Sub
Private Sub cmdOpenCaseFrm_Click()
On Error GoTo Err_cmdOpenCaseFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "CASE"
    'stDocName1 = "DEFENDANT"

    stLinkCriteria = "[CaseKey]=" & Me![CaseKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria


    'Close form
    DoCmd.Close acForm, "DEFENDANT", acSaveYes


Exit_cmdOpenCaseFrm_Click:
    Exit Sub

Err_cmdOpenCaseFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCaseFrm_Click

End Sub

Private Sub Weapon_AfterUpdate()

'If Weapon seized is true set cooperator page visible
If (Me!Weapon = True) Then
    Me!tpgFIREARMS.Visible = True

ElseIf (Me!Weapon = False) Then
    Me!tpgFIREARMS.Visible = False
Else
End If

End Sub

Private Sub Weapon_Click()

End Sub
Private Sub cmdPrintRpt_Click()
On Error GoTo Err_cmdPrintRpt_Click

    Dim stDocName As String

    stDocName = "DEFENDANT"
    DoCmd.OpenReport stDocName, acNormal

Exit_cmdPrintRpt_Click:
    Exit Sub

Err_cmdPrintRpt_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrintRpt_Click

End Sub
Private Sub cmdPhoto_Click()
On Error GoTo Err_cmdPhoto_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmDEF_PHOTOS"

    stLinkCriteria = "[DefKey]=" & Me![DefKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

    'do not allow records to be edited
    Forms![frmDEF_PHOTOS].oleMugA.Locked = True
    Forms![frmDEF_PHOTOS].oleMugB.Locked = True

Exit_cmdPhoto_Click:
    Exit Sub

Err_cmdPhoto_Click:
    MsgBox Err.Description
    Resume Exit_cmdPhoto_Click

End Sub
Private Sub cmdEnterPhoto_Click()
On Error GoTo Err_cmdEnterPhoto_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmDEF_PHOTOS"

    stLinkCriteria = "[DefKey]=" & Me![DefKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

    'Allow records to be edited
    Forms![frmDEF_PHOTOS].oleMugA.Locked = False
    Forms![frmDEF_PHOTOS].oleMugB.Locked = False

Exit_cmdEnterPhoto_Click:
    Exit Sub

Err_cmdEnterPhoto_Click:
    MsgBox Err.Description
    Resume Exit_cmdEnterPhoto_Click

```

```
400  End Sub
```

Form: DEFENDANTList

Code

```
Attribute VB_Name = "Form_DEFENDANTList"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit


Private Sub Form_Current()

'Enable or disable Details button
If IsNull(Me![DefKey]) Then
    Me!cmdOpenDefFrm.Enabled = False
Else
    Me!cmdOpenDefFrm.Enabled = True
End If

'Enable or disable PTS button
If IsNull(Me!USMSNum) Then
    Me!cmdSearchPTS.Enabled = True
Else
    Me!cmdSearchPTS.Enabled = False
End If
End Sub

Private Sub Form_Load()

'Enable or disable Details button
If IsNull(Me![DefKey]) Then
    Me!cmdOpenDefFrm.Enabled = False
Else
    Me!cmdOpenDefFrm.Enabled = True
End If

End Sub

Private Sub Form_Open(Cancel As Integer)

End Sub
Private Sub cmdOpenDefFrm_Click()
On Error GoTo Err_cmdOpenDefFrm_Click


    Dim stDocName As String
    Dim stLinkCriteria As String
    Dim stTitle As String
    Dim stMsg As String
    Dim stMsgTrans As String
    Dim stTitleTrans As String
    Dim IndctNum As Variant
    Dim TopChg As Variant

    stDocName = "DEFENDANT"


    stLinkCriteria = "[DefKey]=" & Me![DefKey]

    'Save record first
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70

    'Check to see if there is a valid record selected before opening form
    'If IsNull(Me![DefKey]) Then 'Notify user and then exit sub
    '    stMsg = "You must be at a valid record before you can select this option"
    '    stTitle = "Select a Record and then Continue"
    '        If MsgBox(stMsg, vbOKOnly, stTitle) = vbOK Then
    '            Exit Sub
    '        End If
    'End If

    'get incictment# and top charge for use in transfer
    IndctNum = Forms![CASE]!IndictmentNum.Value
    TopChg = Forms![CASE]!StatuteNum.Value


    DoCmd.OpenForm stDocName, , , stLinkCriteria

    'Set focus to Middle Name
    Forms![Defendant]!DefMName.SetFocus

    'Disable all FindBy options
    Forms![Defendant]!grpFindOptions.Enabled = False
    Forms![Defendant]!cboFindName.Enabled = False
    Forms![Defendant]!cmdSubmit.Enabled = False
    Forms![Defendant]!txtFindBy.Enabled = False
    'Remove navigation buttons
    Forms![Defendant]!cmdPrevious.Visible = False
    Forms![Defendant]!cmdNext.Visible = False

    'Check if Case Instrument# & TopCharge should be entered for defendant
    'If IsNull(Forms!Defendant!tpgStatutes.Form.Controls!Count) And
IsNull(Forms!Defendant!tpgStatutes.Form.Controls!StatuteNum) Then
    '    stMsgTrans = "Enter Case Instrument no. " & IndctNum & " and Top Charge " &
TopChg & " for Defendant?"
    '    stTitleTrans = "RACKETS"
    '    If MsgBox(stMsgTrans, vbYesNo, stTitleTrans) = vbYes Then
    '        Forms![Defendant]![STATUTESsubform].Count.Value = 1

    '        Forms![Defendant]![STATUTESsubform].StatuteNum.Value = TopChg
    '    Else
    '    End If
    'End If

Exit_cmdOpenDefFrm_Click:
    Exit Sub

Err_cmdOpenDefFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenDefFrm_Click

End Sub
Private Sub cmdSearchPTS_Click()
On Error GoTo Err_cmdSearchPTS_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "CurrentPrisoners"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdSearchPTS_Click:
    Exit Sub

Err_cmdSearchPTS_Click:
    MsgBox Err.Description
    Resume Exit_cmdSearchPTS_Click

End Sub
```

Form: EVENTsubform

Code

```
Attribute VB_Name = "Form_EVENTsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit



Private Sub Form_AfterInsert()

Dim strMessage As String
Dim strLinkCriteria As String

strMessage = "Enter Proffer information now? At the very least, a topic must be entered to continue."

If Me.event_type = "Proffer" Then
        If Forms![Defendant].Coop.Value = False Then
            Forms![Defendant].Coop.Value = True
            Forms![Defendant]!tpgCooperator.Visible = True
        End If
        Forms![Defendant]!subCooperator.Form.Status = "PROF"
    'If Confirm(strMessage) Then
        Forms![Defendant]!tpgCooperator.SetFocus
        Forms![Defendant]!subCooperator.Form.Topic.SetFocus
    'Else
    'End If
End If

End Sub
```

Code

```
Attribute VB_Name = "Form_FIREARMSsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub FirearmType_NotInList(NewData As String, Response As Integer)

Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of Firearm Types?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("firearm_type")
    rstTypes.AddNew
    rstTypes!FirearmType = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If


End Sub

Private Sub Make_NotInList(NewData As String, Response As Integer)

Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of Firearm Makes?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("firearm_make")
    rstTypes.AddNew
    rstTypes!Make = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If


End Sub

Private Sub Model_NotInList(NewData As String, Response As Integer)

Dim strMessage As String
Dim dbsRkts As Database
Dim rstTypes As Recordset

strMessage = "Add '" & NewData & "to the list of Firearm Models?"

If Confirm(strMessage) Then

    'open table and add record
    Set dbsRkts = CurrentDb()
    Set rstTypes = dbsRkts.OpenRecordset("firearm_model")
    rstTypes.AddNew
    rstTypes!Model = NewData
    rstTypes.Update
    Response = acDataErrAdded   'requery list
Else
    Response = acDataErrDisplay ' display error
End If


End Sub
```

Form: frmDEF_PHOTOS

Code

```
Attribute VB_Name = "Form_frmDEF_PHOTOS"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click


    DoCmd.PrintOut

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click

    'save record first
    DoCmd.DoMenuItem acFormBar, acRecordsMenu, acSaveRecord, , acMenuVer70

    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
```

Form: frmDefsByAUSA

Code

```
Attribute VB_Name = "Form_frmDefsByAUSA"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdPreview_Click()
On Error GoTo Err_cmdPreview_Click

    Dim stDocName As String

    stDocName = "rptDefsByAUSA"
    DoCmd.OpenReport stDocName, acPreview
    DoCmd.Close acForm, "frmDefsByAUSA"

Exit_cmdPreview_Click:
    Exit Sub

Err_cmdPreview_Click:
    MsgBox Err.Description
    Resume Exit_cmdPreview_Click

End Sub
Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click

    Dim stDocName As String

    stDocName = "rptDefsByAUSA"
    DoCmd.OpenReport stDocName, acNormal
    DoCmd.Close acForm, "frmDefsByAUSA"

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub
Private Sub cmdCancel_Click()
On Error GoTo Err_cmdCancel_Click


    DoCmd.Close

Exit_cmdCancel_Click:
    Exit Sub

Err_cmdCancel_Click:
    MsgBox Err.Description
    Resume Exit_cmdCancel_Click

End Sub
```

Form: frmDefsByUnit

Code

```
Attribute VB_Name = "Form_frmDefsByUnit"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdPreview_Click()
On Error GoTo Err_cmdPreview_Click

    Dim stDocName As String

    stDocName = "rptDefsByUnit"
    DoCmd.OpenReport stDocName, acPreview
    DoCmd.Close acForm, "frmDefsByUnit"

Exit_cmdPreview_Click:
    Exit Sub

Err_cmdPreview_Click:
    MsgBox Err.Description
    Resume Exit_cmdPreview_Click

End Sub
Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click

    Dim stDocName As String

    stDocName = "rptDefsByUnit"
    DoCmd.OpenReport stDocName, acNormal
    DoCmd.Close acForm, "frmDefsByUnit"

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub
Private Sub cmdCancel_Click()
On Error GoTo Err_cmdCancel_Click


    DoCmd.Close

Exit_cmdCancel_Click:
    Exit Sub

Err_cmdCancel_Click:
    MsgBox Err.Description
    Resume Exit_cmdCancel_Click

End Sub
```

Form: frmFindAlias

Code

```
Attribute VB_Name = "Form_frmFindAlias"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClear_Click()

Dim strSql As String
Dim strFullSql As String

Me!txtAlias = ""

strSql = "WHERE FALSE"
strFullSql = "Select * From qFindAlias " & strSql
Me!subFindAlias.Form.RecordSource = strFullSql

strFullSql = "Select * From qFindAKA " & strSql
Me!subFindAKA.Form.RecordSource = strFullSql

End Sub

Private Sub cmdSubmit_Click()

Dim strAlias As String
Dim strAliasSql As String
Dim strFullSql As String
Dim strMsg As String
Dim strTitle As String
Dim strWhere As String

If IsNull(Me!txtAlias) Or Me!txtAlias = "" Then
    strMsg = "Query is empty. Enter an alias and try again."
    strTitle = "Empty Query!"
        If MsgBox(strMsg, vbOKOnly, strTitle) = vbOK Then
           Exit Sub
        End If
End If

strWhere = "Where "

Select Case grpDatabase
'Search PTS
Case 1
    If grpMatch.Value = 1 Then
        strAlias = "([name_fname] = " & "Forms!frmFindAlias![txtAlias]OR[name_lname] =
" & "Forms!frmFindAlias![txtAlias]OR[name_mname] = " &
"Forms!frmFindAlias![txtAlias]OR[rm_desc] = " & "Forms!frmFindAlias![txtAlias])"

        strAliasSql = strWhere & strAlias
        strFullSql = "Select * From qFindAKA " & strAliasSql
        Me!subFindAKA.Form.RecordSource = strFullSql
    Else
        strAliasSql = " Where [name_fname] Like '" & Forms!frmFindAlias![txtAlias] &
"*' or [name_lname] Like '" & Forms!frmFindAlias![txtAlias] & "*' or [name_mname] Like
'" & Forms!frmFindAlias![txtAlias] & "*' or [rm_desc] Like '*" &
Forms!frmFindAlias![txtAlias] & "*'"

        strFullSql = "Select * From qFindAKA " & strAliasSql
        Me!subFindAKA.Form.RecordSource = strFullSql
    End If

```

```
'Search Rackets
```

```
Case 2
    If grpMatch.Value = 1 Then
        strAlias = "([Alias] = " & "Forms!frmFindAlias![txtAlias]OR[AliasLN] = " &
"Forms!frmFindAlias![txtAlias])"
        strAliasSql = strWhere & strAlias
        strFullSql = "Select * From qFindAlias " & strAliasSql
        Me!subFindAlias.Form.RecordSource = strFullSql
    Else
        strAliasSql = " Where [Alias] Like '" & Forms!frmFindAlias![txtAlias] & "*' or
[AliasLN] Like '" & Forms!frmFindAlias![txtAlias] & "*'"
        strFullSql = "Select * From qFindAlias " & strAliasSql
        Me!subFindAlias.Form.RecordSource = strFullSql
    End If
End Select

End Sub

Private Sub Form_AfterUpdate()

End Sub

Private Sub Form_Load()

End Sub

Private Sub Frame4_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub grpDatabase_AfterUpdate()

'Set subform visible based on selection of search
Select Case grpDatabase

Case 1

Me!subFindAKA.Form.Visible = True
Me!subFindAlias.Form.Visible = False

Case 2
Me!subFindAKA.Form.Visible = False
Me!subFindAlias.Form.Visible = True

End Select

End Sub

Private Sub txtAlias_BeforeUpdate(Cancel As Integer)

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
Private Sub cmdOpenDefFrm_Click()
On Error GoTo Err_cmdOpenDefFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

```

Form: frmFindAlias

```
    'stLinkCriteria = Me!subFindAKA.Form.

    stDocName = "DEFENDANT"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenDefFrm_Click:
    Exit Sub

Err_cmdOpenDefFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenDefFrm_Click

End Sub
```

Code

```
Attribute VB_Name = "Form_frmFindAssoc"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClear_Click()
Dim strSql As String
Dim strFull As String

Me!txtFind = ""

strSql = "Where False"
strFull = "Select * From qFindAssoc " & strSql
Me!subFindAssoc.Form.RecordSource = strFull
Me!subFindLocale.Form.RecordSource = strFull
Me!subFindCohort.Form.RecordSource = strFull

End Sub

Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub

Private Sub cmdSubmit_Click()


Dim strAssoc As String
Dim strSql As String
Dim strMsg As String
Dim strTitle As String

'clear variables
'strAssoc = ""
'strSql = ""

'First check to see if a string was entered, if not end routine
strMsg = "Query is empty. Enter a Search String and try again"
strTitle = "RACKETS"
If IsNull(Me!txtFind) Or Me!txtFind = "" Then
    If MsgBox(strMsg, vbOKOnly, strTitle) = vbOK Then
        Exit Sub
    End If
End If

'run query based on selection type
Select Case grpType

'search Group table via qFindAssoc
Case 1
    strAssoc = "Where [GroupName] Like '" & Forms!frmFindAssoc![txtFind] & "*'"
    strSql = "Select * From qFindAssoc " & strAssoc
    Me!subFindAssoc.Form.RecordSource = strSql

'search locale via q FindLocale
Case 2

    strAssoc = "Where [lo_name] Like '" & Forms!frmFindAssoc![txtFind] & "*'"
    strSql = "Select * From qFindLocale " & strAssoc
    Me!subFindLocale.Form.RecordSource = strSql

 'search cohort via qFindCohort
Case 3
    strAssoc = "Where [co_lname] Like '" & Forms!frmFindAssoc![txtFind] & "*' or
[co_lname] Like '" & Forms!frmFindAssoc![txtFind] & "*'"
    strSql = "Select * From qFindCohort " & strAssoc
    Me!subFindCohort.Form.RecordSource = strSql

End Select
End Sub

Private Sub Form_Close()
Me!txtFind = ""
End Sub

Private Sub Form_Open(Cancel As Integer)

Dim strSql As String
Dim strFull As String

'Clear last query
Me!txtFind = ""

strSql = "Where False"
strFull = "Select * From qFindAssoc " & strSql

Me!subFindAssoc.Form.RecordSource = strFull

'Set default visible and all others not visible
Me!subFindAssoc.Form.Visible = True
Me!subFindCohort.Form.Visible = False
Me!subFindLocale.Form.Visible = False


End Sub

Private Sub grpType_AfterUpdate()

'Set the subform based on the query type
Select Case grpType

Case 1
    Me!subFindAssoc.Form.Visible = True
    Me!subFindLocale.Form.Visible = False
    Me!subFindCohort.Form.Visible = False

Case 2
    Me!subFindLocale.Form.Visible = True
    Me!subFindAssoc.Form.Visible = False
    Me!subFindCohort.Form.Visible = False

Case 3
    Me!subFindCohort.Form.Visible = True
    Me!subFindAssoc.Form.Visible = False
    Me!subFindLocale.Form.Visible = False

End Select

End Sub
```

Form: frmFindCounsel

Code

```
Attribute VB_Name = "Form_frmFindCounsel"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cboFind_AfterUpdate()

Dim strSql As String
Dim strFull As String

strSql = "Select * From qFindCounsel Where [Counsel] = " &
"Forms!frmFindCounsel![cboFind]"
Me!subFindCounsel.Form.RecordSource = strSql


End Sub

Private Sub cboFind_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub cboFind_Change()
Dim strSql As String
Dim strFull As String

'strSql = "Select * From qFindCounsel Where [] = " & "Forms!frmFindCounsel![cboFind]"

'Me!subFindCounsel.Form.RecordSource = strSql


End Sub

Private Sub Detail_Click()

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
```

Form: frmFindEmployer

Code

```
Attribute VB_Name = "Form_frmFindEmployer"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClear_Click()

Dim strSql As String
Dim strFull As String

Me!txtFind = ""

strSql = "Where False"
strFull = "Select * From qFindEmployer " & strSql
Me!subFindEmployer.Form.RecordSource = strFull
Me!subFindEmployerPTS.Form.RecordSource = strFull

End Sub

Private Sub cmdSubmit_Click()

Dim strEmploy As String
Dim strSql As String
Dim strMsg As String
Dim strTitle As String


'First check to see if a string was entered, if not end routine
strMsg = "Query is empty. Enter a Search String and try again"
strTitle = "RACKETS"
If IsNull(Me!txtFind) Or Me!txtFind = "" Then
    If MsgBox(strMsg, vbOKOnly, strTitle) = vbOK Then
        Exit Sub
    End If
End If

'run query based on selection type
Select Case grpType

'search by employer
Case 1

    If grpSearch.Value = 1 Then
        strEmploy = "Where [em_name] Like '" & Forms!frmFindEmployer![txtFind] & "*'"
        strSql = "Select * From qFindEmployer " & strEmploy
        Me!subFindEmployer.Form.RecordSource = strSql
    Else
        strEmploy = "Where [em_name] Like '" & Forms!frmFindEmployer![txtFind] & "*'"
        strSql = "Select * From qFindEmployPTS " & strEmploy
        Me!subFindEmployerPTS.Form.RecordSource = strSql
    End If

'search by owner/supervisor
Case 2
    If grpSearch.Value = 1 Then
        strEmploy = "Where [em_poc] Like '*" & Forms!frmFindEmployer![txtFind] & "*'"
        strSql = "Select * From qFindEmployer " & strEmploy
        Me!subFindEmployer.Form.RecordSource = strSql
    Else
        strEmploy = "Where [em_poc] Like '*" & Forms!frmFindEmployer![txtFind] & "*'"
        strSql = "Select * From qFindEmployPTS " & strEmploy
        Me!subFindEmployerPTS.Form.RecordSource = strSql
    End If

'search by occupation
Case 3
    If grpSearch.Value = 1 Then
        strEmploy = "Where [em_occ] Like '" & Forms!frmFindEmployer![txtFind] & "*'"
        strSql = "Select * From qFindEmployer " & strEmploy
        Me!subFindEmployer.Form.RecordSource = strSql
    Else
        strEmploy = "Where [em_occ] Like '" & Forms!frmFindEmployer![txtFind] & "*'"
        strSql = "Select * From qFindEmployPTS " & strEmploy
        Me!subFindEmployerPTS.Form.RecordSource = strSql
    End If

End Select


End Sub

Private Sub Form_Open(Cancel As Integer)

Me!subFindEmployer.Form.Visible = True
Me!subFindEmployerPTS.Form.Visible = False

End Sub

Private Sub grpSearch_AfterUpdate()

'Set subform based on search selection
Select Case grpSearch

Case 1
'Search records linked to defendants in Rackets
Me!subFindEmployer.Form.Visible = True
Me!subFindEmployerPTS.Form.Visible = False

Case 2
'Search records not linked to an active defendant
Me!subFindEmployer.Form.Visible = False
Me!subFindEmployerPTS.Form.Visible = True

End Select
End Sub

Private Sub grpSearch_BeforeUpdate(Cancel As Integer)

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
```

Form: frmFindNarcs

Code

```
Attribute VB_Name = "Form_frmFindNarcs"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClear_Click()

Dim strSql As String
Dim strFullSql As String
Me!txtFind = ""
Me!cboType = ""

strSql = "Where False"
strFullSql = "Select * From qFindNarc " & strSql
Me!subFindNarc.Form.RecordSource = strFullSql

End Sub

Private Sub cmdSubmit_Click()

Dim strType As String
Dim strNarc As String
Dim strBoth As String
Dim strSql As String
Dim strMsg As String
Dim strTitle As String



Select Case grpSearch

Case 1
    strType = "Where [NarcType]= " & "Forms!frmFindNarcs![cboType]"
    strSql = "Select * From qFindNarc " & strType
    Me!subFindNarc.Form.RecordSource = strSql

Case 2
    strNarc = "Where [Brand]= " & "Forms!frmFindNarcs![txtFind]"
    strSql = "Select * From qFindNarc " & strNarc
    Me!subFindNarc.Form.RecordSource = strSql


Case 3
    strType = "[NarcType]= " & "Forms!frmFindNarcs![cboType]"
    strNarc = "[Brand]= " & "Forms!frmFindNarcs![txtFind]"
    strBoth = "Where " & strType & " AND " & strNarc
    strSql = "Select * From qFindNarc " & strBoth
    Me!subFindNarc.Form.RecordSource = strSql


End Select

End Sub

Private Sub Detail_Click()

End Sub
Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close
```

```

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub

Private Sub Form_Open(Cancel As Integer)


Select Case grpSearch

Case 1
    cboType.BackColor = 16777215
    txtFind.BackColor = 12632256
    Me.cboType.SetFocus
Case 2
    txtFind.BackColor = 16777215
    cboType.BackColor = 12632256
    Me.txtFind.SetFocus
Case 3
    cboType.BackColor = 16777215
    txtFind.BackColor = 16777215
    Me.cboType.SetFocus
End Select

End Sub

Private Sub grpSearch_AfterUpdate()
Select Case grpSearch

Case 1
    cboType.BackColor = 16777215
    txtFind.BackColor = 12632256
    Me.cboType.SetFocus
    Me!txtFind = ""

Case 2
    txtFind.BackColor = 16777215
    cboType.BackColor = 12632256
    Me.txtFind.SetFocus
    Me!cboType = ""

Case 3
    cboType.BackColor = 16777215
    txtFind.BackColor = 16777215
    Me.cboType.SetFocus
End Select

End Sub

Private Sub grpSearch_BeforeUpdate(Cancel As Integer)

End Sub
```

Form: frmPrisDetails

Code

```
Attribute VB_Name = "Form_frmPrisDetails"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdPreview_Click()
On Error GoTo Err_cmdPreview_Click

    Dim stDocName As String

    stDocName = "rptPrisonerDetails"
    DoCmd.OpenReport stDocName, acPreview

    DoCmd.Close acForm, "frmPrisDetails", acSaveNo

Exit_cmdPreview_Click:
    Exit Sub

Err_cmdPreview_Click:
    MsgBox Err.Description
    Resume Exit_cmdPreview_Click

End Sub

Private Sub cmdSetDate_Click()
'set date equal to date on calender
txtDate = dwnldDate.Value

End Sub

Private Sub dwnldDate_Updated(Code As Integer)

End Sub

Private Sub Form_Current()

End Sub

Private Sub Form_Load()
'set controls equal to today
Me!dwnldDate = Date
Me!txtDate = Date

End Sub
```

Form: frmPrisDetails

```
Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click

    Dim stDocName As String

    stDocName = "rptPrisonerDetails"
    DoCmd.OpenReport stDocName, acNormal

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub
Private Sub cmdCancel_Click()
On Error GoTo Err_cmdCancel_Click


    DoCmd.Close

Exit_cmdCancel_Click:
    Exit Sub

Err_cmdCancel_Click:
    MsgBox Err.Description
    Resume Exit_cmdCancel_Click

End Sub
```

Code

```
Attribute VB_Name = "Form_frmPTS"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub

Private Sub cmdFindUSMS_Click()
Dim rst As Recordset

    Dim strCriteria As String
    Dim strUSMS As String

        'Make sure there is an entry in the text box before running
        If IsNull(txtFindUSMS) Then
            MsgBox "Enter a USMS numer first and then submit query"
            Exit Sub
        End If



        Set rst = Me.RecordsetClone
        strUSMS = Me!txtFindUSMS
        strCriteria = "[pr_usms_no]='" & strUSMS & "'"
        rst.FindFirst strCriteria
        If rst.NoMatch Then
            MsgBox "The number you entered was not found."
        Else
        Me.Bookmark = Me.RecordsetClone.Bookmark
        End If
        Me!txtFindUSMS.Value = ""

End Sub

Private Sub cmdNext_Click()
On Error GoTo Err_cmdNext_Click


    DoCmd.GoToRecord , , acNext

Exit_cmdNext_Click:
    Exit Sub

Err_cmdNext_Click:
    MsgBox Err.Description
    Resume Exit_cmdNext_Click

End Sub
Private Sub cmdPrevious_Click()
```

```
On Error GoTo Err_cmdPrevious_Click


    DoCmd.GoToRecord , , acPrevious

Exit_cmdPrevious_Click:
    Exit Sub

Err_cmdPrevious_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrevious_Click

End Sub
Sub Combo33_AfterUpdate()

    ' Find the record that matches the control.
    If IsNull(Me![Combo33]) Then
        Exit Sub
    End If

    ' Find the record that matches the control.
    Me.RecordsetClone.FindFirst "[pr_key] = " & Me![Combo33]
    Me.Bookmark = Me.RecordsetClone.Bookmark
End Sub

Private Sub Form_Current()

'synchronize combo box with record
Combo33 = pr_key
End Sub
Private Sub cmdPrintReport_Click()
On Error GoTo Err_cmdPrintReport_Click

    Dim stDocName As String

    stDocName = "rptPrisonerDetailsPTS"
    DoCmd.OpenReport stDocName, acNormal

Exit_cmdPrintReport_Click:
    Exit Sub

Err_cmdPrintReport_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrintReport_Click

End Sub
```

Form: frmPTSArrests

Code

```
Attribute VB_Name = "Form_frmPTSArrests"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit
Const ConErrRptCanceled = 2501


Private Sub cmdCancel_Click()

' Close form
On Error GoTo Err_cmdCancel_Click

    DoCmd.Close

Exit_cmdCancel_Click:
    Exit Sub

Err_cmdCancel_Click:
    MsgBox Err.Description
    Resume Exit_cmdCancel_Click

End Sub



Private Sub cmdPreview_Click()

' Preview report.
Dim strDocName As String

    On Error GoTo Err_cmdPreview_Click

    ' Check to see that ending date is later than beginning date.
    If IsDate(BeginDate) And IsDate(EndDate) Then
        If EndDate < BeginDate Then
            MsgBox "The end date must be later than the begin date."
            cmdSetDate.Caption = "Set End Date"
            SelectDate.SetFocus
            Exit Sub
        End If
    Else
        MsgBox "Please use a valid date for the beginning date and the ending date values."

Exit Sub

    End If

    strDocName = "PTS Arrests"
    DoCmd.OpenReport strDocName, acViewPreview

Exit_cmdPreview_Click:
    Exit Sub

Err_cmdPreview_Click:
    If Err = ConErrRptCanceled Then
        Resume Exit_cmdPreview_Click
    Else
        MsgBox Err.Description
        Resume Exit_cmdPreview_Click
    End If

End Sub
```

Form: frmPTSArrests

```

Private Sub cmdPrint_Click()
    ' Print report.
    Dim strDocName As String

    On Error GoTo Err_cmdPrint_Click

    ' Check to see that ending date is later than beginning date.
    If EndDate < BeginDate Then
        MsgBox "The end date must be later than the begin date."
        cmdSetDate.Caption = "Set End Date"
        SelectDate.SetFocus
        Exit Sub
    End If

    strDocName = "PTS Arrests"
    DoCmd.OpenReport strDocName, acViewNormal

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    If Err = ConErrRptCanceled Then
        Resume Exit_cmdPrint_Click
    Else
        MsgBox Err.Description
        Resume Exit_cmdPrint_Click
    End If

End Sub


Private Sub cmdSetDate_Click()

    On Error GoTo cmdSetDate_Error

    If cmdSetDate.Caption = "Set Begin Date" Then
        BeginDate = SelectDate.Value
        cmdSetDate.Caption = "Set End Date"
    Else
        EndDate = SelectDate.Value
        cmdSetDate.Caption = "Set Begin Date"
    End If

    Exit Sub

cmdSetDate_Error:
    MsgBox Err.Description
    Exit Sub

End Sub

Function IsLoaded(strfrmName As String) As Integer

    ' Determines if a form is loaded.

    Const conFormDesign = 0
    Dim intX As Integer

    IsLoaded = False
    For intX = 0 To Forms.Count - 1
        If Forms(intX).FormName = strfrmName Then
            If Forms(intX).CurrentView <> conFormDesign Then
                IsLoaded = True
                Exit Function       ' Quit function once form has been found.
            End If
        End If
    Next

```

Form: frmPTSArrests

```
End Function


Private Sub Detail_Click()

End Sub

Private Sub Form_Load()

'Enter parameters in date boxes
Me!SelectDate = Now()
Me!BeginDate = Now() - 7
Me!EndDate = Now()

End Sub
```

Form: frmREPORTS

Code

```
Attribute VB_Name = "Form_frmREPORTS"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdOpenRptPtsDwn_Click()
On Error GoTo Err_cmdOpenRptPtsDwn_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "PTSDownloads"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenRptPtsDwn_Click:
    Exit Sub

Err_cmdOpenRptPtsDwn_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenRptPtsDwn_Click

End Sub
```

Form: frmTARGET_PHOTOsub

Code

```
Attribute VB_Name = "Form_frmTARGET_PHOTOsub"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdClose_Click()
On Error GoTo Err_cmdClose_Click


    DoCmd.Close

Exit_cmdClose_Click:
    Exit Sub

Err_cmdClose_Click:
    MsgBox Err.Description
    Resume Exit_cmdClose_Click

End Sub
Private Sub cmdPrint_Click()
On Error GoTo Err_cmdPrint_Click


    DoCmd.PrintOut

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    MsgBox Err.Description
    Resume Exit_cmdPrint_Click

End Sub
```

Form: frmTARGETS

Code

```
Attribute VB_Name = "Form_frmTARGETS"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdPhoto_Click()
On Error GoTo Err_cmdPhoto_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmTARGET_PHOTOsub"

    stLinkCriteria = "[TargetKey]=" & Me![TargetKey]

    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdPhoto_Click:
    Exit Sub

Err_cmdPhoto_Click:
    MsgBox Err.Description
    Resume Exit_cmdPhoto_Click

End Sub

Private Sub Form_AfterUpdate()

'enable or disable photo buttons based on record status
    If IsNull(Me![TargetKey]) And IsNull(Me![txtPhoto]) Then
        Me!cmdPhoto.Enabled = False
    Else
        Me!cmdPhoto.Enabled = True
    End If

    If Not (IsNull(Me![TargetKey])) And Not (IsNull(Me![txtPhoto])) Then
        Me!cmdInsertPhoto.Enabled = True
    Else
        Me!cmdInsertPhoto.Enabled = False
    End If
'enable or disable photo buttons based on record status
    If IsNull(Me![TargetKey]) And IsNull(Me![txtPhoto]) Then
        Me!cmdPhoto.Enabled = False
    Else

        Me!cmdPhoto.Enabled = True
    End If

    If Not (IsNull(Me![TargetKey])) And Not (IsNull(Me![txtPhoto])) Then
        Me!cmdInsertPhoto.Enabled = True
    Else
        Me!cmdInsertPhoto.Enabled = False
    End If

End Sub

Private Sub Form_Current()

'enable or disable photo buttons based on record status
    If IsNull(Me![TargetKey]) Or IsNull(Me![txtPhoto]) Then
        Me!cmdPhoto.Enabled = False
    Else
        Me!cmdPhoto.Enabled = True
    End If

    If Not (IsNull(Me![TargetKey])) And (IsNull(Me![txtPhoto])) Then
        Me!cmdInsertPhoto.Enabled = True
    Else
        Me!cmdInsertPhoto.Enabled = False
    End If

End Sub

Private Sub Form_Load()
    If IsNull(Me![TargetKey]) Then
        Me!cmdPhoto.Enabled = False
    Else
        Me!cmdPhoto.Enabled = True
    End If

End Sub
Private Sub cmdInsertPhoto_Click()
On Error GoTo Err_cmdInsertPhoto_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmTARGET_PHOTOsub"

    stLinkCriteria = "[TargetKey]=" & Me![TargetKey]
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdInsertPhoto_Click:
    Exit Sub

Err_cmdInsertPhoto_Click:

    MsgBox Err.Description
    Resume Exit_cmdInsertPhoto_Click

End Sub
```

Form: HOMICIDEsubform

Code

```
Attribute VB_Name = "Form_HOMICIDEsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Form_Open(Cancel As Integer)

'On opening set focus to Homicide date
Me!HomDate.SetFocus

End Sub
```

Form: LOCALEsubform

Code

```
Attribute VB_Name = "Form_LOCALEsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Form_Load()

Me!LocName.SetFocus
Me.Repaint

End Sub
```

Form: NARCOTICSsubform

Code

```
Attribute VB_Name = "Form_NARCOTICSsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Form_Current()

End Sub

Private Sub NarcType_AfterUpdate()

If IsNull(Me!NarcType) Then
    Forms!COOPERATORS!subNARCOTICS!subNARC_BRAND!Brand.Enabled = False
Else
    Forms!COOPERATORS!subNARCOTICS!subNARC_BRAND!Brand.Enabled = True
End If

End Sub

Private Sub NarcType_BeforeUpdate(Cancel As Integer)

End Sub
```

Code

```
Attribute VB_Name = "Form_PTSDownloads"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit
Const ConErrRptCanceled = 2501


Private Sub cmdCancel_Click()

' Close form
On Error GoTo Err_cmdCancel_Click

    DoCmd.Close

Exit_cmdCancel_Click:
    Exit Sub

Err_cmdCancel_Click:
    MsgBox Err.Description
    Resume Exit_cmdCancel_Click

End Sub



Private Sub cmdPreview_Click()

' Preview report.
Dim strDocName As String

    On Error GoTo Err_cmdPreview_Click

    ' Check to see that ending date is later than beginning date.
    If IsDate(BeginDate) And IsDate(EndDate) Then
        If EndDate < BeginDate Then
            MsgBox "The end date must be later than the begin date."
            cmdSetDate.Caption = "Set End Date"
            SelectDate.SetFocus
            Exit Sub
        End If
    Else
        MsgBox "Please use a valid date for the beginning date and the ending date values."

Exit Sub

    End If

    strDocName = "PTS Downloads"
    DoCmd.OpenReport strDocName, acViewPreview

Exit_cmdPreview_Click:
    Exit Sub

Err_cmdPreview_Click:
    If Err = ConErrRptCanceled Then
        Resume Exit_cmdPreview_Click
    Else
        MsgBox Err.Description
        Resume Exit_cmdPreview_Click
    End If

End Sub
```

Form: PTSDownloads

```

Private Sub cmdPrint_Click()
    ' Print report.
    Dim strDocName As String

    On Error GoTo Err_cmdPrint_Click

    ' Check to see that ending date is later than beginning date.
    If EndDate < BeginDate Then
        MsgBox "The end date must be later than the begin date."
        cmdSetDate.Caption = "Set End Date"
        SelectDate.SetFocus
        Exit Sub
    End If

    strDocName = "PTS Downloads"
    DoCmd.OpenReport strDocName, acViewNormal

Exit_cmdPrint_Click:
    Exit Sub

Err_cmdPrint_Click:
    If Err = ConErrRptCanceled Then
        Resume Exit_cmdPrint_Click
    Else
        MsgBox Err.Description
        Resume Exit_cmdPrint_Click
    End If

End Sub


Private Sub cmdSetDate_Click()

    On Error GoTo cmdSetDate_Error

    If cmdSetDate.Caption = "Set Begin Date" Then
        BeginDate = SelectDate.Value
        cmdSetDate.Caption = "Set End Date"
    Else
        EndDate = SelectDate.Value
        cmdSetDate.Caption = "Set Begin Date"
    End If

    Exit Sub

cmdSetDate_Error:
    MsgBox Err.Description
    Exit Sub

End Sub

Function IsLoaded(strfrmName As String) As Integer

    ' Determines if a form is loaded.

    Const conFormDesign = 0
    Dim intX As Integer

    IsLoaded = False
    For intX = 0 To Forms.Count - 1
        If Forms(intX).FormName = strfrmName Then
            If Forms(intX).CurrentView <> conFormDesign Then
                IsLoaded = True
                Exit Function       ' Quit function once form has been found.
            End If
        End If
    Next

```

```
End Function


Private Sub Detail_Click()

End Sub

Private Sub Form_Load()

'Enter parameters in date boxes
Me!SelectDate = Now()
Me!BeginDate = Now() - 7
Me!EndDate = Now()

End Sub
```

Form: StartUp

Code

```
Attribute VB_Name = "Form_StartUp"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub cmdOpenAlias_Click()
On Error GoTo Err_cmdOpenAlias_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmFindAlias"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenAlias_Click:
    Exit Sub

Err_cmdOpenAlias_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenAlias_Click

End Sub

Private Sub cmdOpenAssoc_Click()
On Error GoTo Err_cmdOpenAssoc_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmFindAssoc"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenAssoc_Click:
    Exit Sub

Err_cmdOpenAssoc_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenAssoc_Click

End Sub

Private Sub cmdOpenCoopFrm_Click()
On Error GoTo Err_cmdOpenCoopFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "COOPERATORS"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenCoopFrm_Click:
    Exit Sub

Err_cmdOpenCoopFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCoopFrm_Click


End Sub
```

```

Private Sub cmdOpenCounsel_Click()
On Error GoTo Err_cmdOpenCounsel_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmFindCounsel"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenCounsel_Click:
    Exit Sub

Err_cmdOpenCounsel_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCounsel_Click

End Sub

Private Sub cmdOpenDefFrm_Click()
On Error GoTo Err_cmdOpenDefFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "DEFENDANT"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenDefFrm_Click:
    Exit Sub

Err_cmdOpenDefFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenDefFrm_Click


End Sub

Private Sub cmdOpenFindEmploy_Click()

End Sub

Private Sub cmdOpenMaint_Click()

Me.GoToPage 4

End Sub

Private Sub cmdOpenNarc_Click()
On Error GoTo Err_cmdOpenNarc_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmFindNarcs"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenNarc_Click:
    Exit Sub

Err_cmdOpenNarc_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenNarc_Click

End Sub
```

```
Private Sub cmdOpenPg3_Click()

Me.GoToPage 3

End Sub

Private Sub cmdOpenRptfrm_Click()
Me.GoToPage 2
End Sub

Private Sub cmdOpenTQfrm_Click()

If Me!lblMain.Visible = True Then
    Me.lblMain.Visible = False
    Me.lblRpt.Visible = False
    Me.lblSQ.Visible = False
    Me.lblTQ.Visible = False
    Me.lblRecords.Visible = False
    Me.lblMaint.Visible = False
    Me.cmdOpenMaint.Visible = False
    Me.cmdOpenPg3.Visible = False
    Me.cmdOpenSQfrm.Visible = False
    Me.cmdOpenRptfrm.Visible = False
    Me.lblAlias.Visible = True
    Me.lblNarc.Visible = True
    Me.lblAssoc.Visible = True
    Me.lblCounsel.Visible = True
    Me.lblEmploy.Visible = True
    Me.lblReturnMain.Visible = True
    Me.cmdOpenAlias.Visible = True
    Me.cmdOpenNarc.Visible = True
    Me.cmdOpenAssoc.Visible = True
    Me.cmdOpenCounsel.Visible = True
    Me.cmdOpenEmploy.Visible = True
    Me.cmdOpenAlias.SetFocus
    Me.cmdOpenTQfrm.Picture = "c:\Program
Files\DevStudio\VB\Graphics\Icons\Arrows\Arw09lt.ico"

ElseIf Me!lblMain.Visible = False Then
    Me.lblMain.Visible = True
    Me.lblRpt.Visible = True
    Me.lblSQ.Visible = True
    Me.lblTQ.Visible = True
    Me.lblRecords.Visible = True
    Me.cmdOpenPg3.Visible = True
    Me.cmdOpenSQfrm.Visible = True
    Me.cmdOpenRptfrm.Visible = True
    Me.lblMaint.Visible = True
    Me.cmdOpenMaint.Visible = True
    Me.lblAlias.Visible = False
    Me.lblNarc.Visible = False
    Me.lblAssoc.Visible = False
    Me.lblCounsel.Visible = False
    Me.lblEmploy.Visible = False
    Me.lblReturnMain.Visible = False
    Me.cmdOpenAlias.Visible = False
    Me.cmdOpenNarc.Visible = False
    Me.cmdOpenAssoc.Visible = False
    Me.cmdOpenCounsel.Visible = False
    Me.cmdOpenEmploy.Visible = False
    Me.cmdOpenTQfrm.Picture = "C:\Program
Files\DevStudio\VB\Graphics\Icons\Misc\LIGHTON.ICO"
End If
End Sub

Private Sub Command30_Click()
```

```
Me.GoToPage 1
End Sub
```

Form: StartUp

```

Private Sub Command34_Click()

End Sub

Private Sub Command40_Click()

Me.GoToPage 1

End Sub

Private Sub Command82_Click()

'run transfer.exe - vb app that inputs email messages into database
Dim RetVal
RetVal = Shell("D:\NETWORK\RACKETS\MISC\TRANSLATE.EXE", 1)

DoCmd.Close acForm, "StartUp"

End Sub

Private Sub Form_Load()
Me.GoToPage 1

Me.lblAlias.Visible = False
Me.lblNarc.Visible = False
Me.lblAssoc.Visible = False
Me.lblCounsel.Visible = False
Me.lblEmploy.Visible = False
Me.lblReturnMain.Visible = False
Me.cmdOpenAlias.Visible = False
Me.cmdOpenNarc.Visible = False
Me.cmdOpenAssoc.Visible = False
Me.cmdOpenCounsel.Visible = False
Me.cmdOpenEmploy.Visible = False
Me.lblMain.Visible = True
Me.lblRpt.Visible = True
Me.lblSQ.Visible = True
Me.lblTQ.Visible = True
Me.lblRecords.Visible = True
Me.lblMaint.Visible = True
Me.cmdOpenMaint.Visible = True
Me.cmdOpenPg3.Visible = True
Me.cmdOpenTQfrm.Visible = True
Me.cmdOpenSQfrm.Visible = True
Me.cmdOpenRptfrm.Visible = True

End Sub

Private Sub Form_Open(Cancel As Integer)

Me.GoToPage 1

Me.lblAlias.Visible = False
Me.lblNarc.Visible = False
Me.lblAssoc.Visible = False
Me.lblCounsel.Visible = False
Me.lblEmploy.Visible = False
Me.lblReturnMain.Visible = False
Me.cmdOpenAlias.Visible = False
Me.cmdOpenNarc.Visible = False
Me.cmdOpenAssoc.Visible = False
Me.cmdOpenCounsel.Visible = False
Me.cmdOpenEmploy.Visible = False
Me.lblMain.Visible = True
Me.lblRpt.Visible = True
Me.lblSQ.Visible = True
Me.lblTQ.Visible = True
Me.lblMaint.Visible = True
```

```
Me.cmdOpenMaint.Visible = True
Me.lblRecords.Visible = True
Me.cmdOpenPg3.Visible = True
Me.cmdOpenTQfrm.Visible = True
Me.cmdOpenSQfrm.Visible = True
Me.cmdOpenRptfrm.Visible = True

End Sub
Private Sub cmdExit_Click()
On Error GoTo Err_cmdExit_Click


    DoCmd.Quit

Exit_cmdExit_Click:
    Exit Sub

Err_cmdExit_Click:
    MsgBox Err.Description
    Resume Exit_cmdExit_Click

End Sub
Private Sub cmdOpenCaseFrm_Click()
On Error GoTo Err_cmdOpenCaseFrm_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "CASE"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenCaseFrm_Click:
    Exit Sub

Err_cmdOpenCaseFrm_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenCaseFrm_Click

End Sub
Private Sub cmdOpenNewCase_Click()
On Error GoTo Err_cmdOpenNewCase_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "CASE"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

    Forms!CASE.DataEntry = True
    Forms!CASE.cmdArchive.Enabled = False
    Forms!CASE.cmdArchive.Visible = False
    Forms!CASE.Combo15.Enabled = False
    Forms!CASE.Combo15.Visible = False
    Forms!CASE.lblUSV.Visible = False

Exit_cmdOpenNewCase_Click:
    Exit Sub

Err_cmdOpenNewCase_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenNewCase_Click

End Sub
```

```
Private Sub cmdOpenPTS_Click()
On Error GoTo Err_cmdOpenPTS_Click

    Dim stDocName As String
```

```
    Dim stLinkCriteria As String

    stDocName = "frmPTS"
    DoCmd.Close acForm, "StartUp"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenPTS_Click:
    Exit Sub

Err_cmdOpenPTS_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenPTS_Click

End Sub
Private Sub cmdOpenEmploy_Click()
On Error GoTo Err_cmdOpenEmploy_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "frmFindEmployer"
    DoCmd.OpenForm stDocName, , , stLinkCriteria
    DoCmd.Close acForm, "StartUp"

Exit_cmdOpenEmploy_Click:
    Exit Sub

Err_cmdOpenEmploy_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenEmploy_Click

End Sub
Private Sub cmdOpenclnArrest_Click()
On Error GoTo Err_cmdOpenclnArrest_Click

    Dim stDocName As String
    Dim stLinkCriteria As String

    stDocName = "cln_ARREST"
    DoCmd.OpenForm stDocName, , , stLinkCriteria

Exit_cmdOpenclnArrest_Click:
    Exit Sub

Err_cmdOpenclnArrest_Click:
    MsgBox Err.Description
    Resume Exit_cmdOpenclnArrest_Click

End Sub
Private Sub cmdOpenPTSArr_Click()
End Sub
```

Form: STATUTESsubform

Code

```
Attribute VB_Name = "Form_STATUTESsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Count_BeforeUpdate(Cancel As Integer)

End Sub

Private Sub Form_AfterUpdate()
End Sub

Private Sub Form_BeforeUpdate(Cancel As Integer)


End Sub

Private Sub Form_Current()


End Sub

Private Sub Form_Error(DataErr As Integer, Response As Integer)

    Const conErrFieldRequired = 3314

    If DataErr = conErrFieldRequired Then
        DisplayMessage "You must enter a Count and Statute number to save the record."
        Response = acDataErrContinue
    Else
        Response = acDataErrDisplay
    End If

End Sub

Private Sub Form_Load()

End Sub

Private Sub Location_AfterUpdate()


End Sub

Private Sub Location_Click()


End Sub

Private Sub StatuteDesc_AfterUpdate()



End Sub
```

Form: WARRANTsubform

Code

```
Attribute VB_Name = "Form_WARRANTsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Form_Open(Cancel As Integer)


End Sub
```

Form: WIRETAPsubform

Code

```
Attribute VB_Name = "Form_WIRETAPsubform"
Attribute VB_Creatable = True
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Compare Database
Option Explicit

Private Sub Form_Open(Cancel As Integer)



End Sub
```