Powered By Nlets
The International Justice and Public Safety Information Sharing Network

AAMVA

National institute of Justice
Grant Number: 2003-RD-CX-K002

CANDLE: Collaboration between AAMVA and Nlets for Driver License Exchange

Final Report

U.S. Department of Justice
Office of Justice Programs
National Institute of Justice

TABLE OF CONTENTS

TABLE OF FIGURES

I. Abstract

Justice and public safety (JPS) professionals in the United States and Canada receive interstate driver and motor vehicle information using Nlets, the International Justice and Public Safety Information Sharing Network. The states presently deliver this key data in non-uniform, state-specific formats. Receipt of this important data must occur in a consistent manner to improve efficiency while reducing public and officer risk.

With sponsorship from the National Institute of Justice (NIJ), Nlets partnered with AAMVA, the American Association of Motor Vehicle Administrators, to leverage AAMVA's interstate driver/vehicle data standards to make this crucial information available to justice and public safety officials in a common, standards-compliant format. Under the NIJ-funded CANDLE grant (*Collaboration between AAMVA and Nlets for Driver License Exchange*), the partner organizations successfully used AAMVA's information model to convey uniform responses over Nlets in a format compliant with the DOJ Global Justice XML Data Model (GJXDM) information sharing standard.

The CANDLE effort has identified the need for common interstate and intrastate exchanges and for a single, shared interface to deliver common services for departments of public safety (DPS) and motor vehicle (DMV). The goal of a single shared interface will be realized by working with AAMVA to extend and promote the CANDLE specifications for adoption and implementation directly by state DMVs. A number of major national initiatives including the Real ID Act will result in major revisions to state DMV systems which create the opportunity for CANDLE DMV implementations. In the absence of the CANDLE support, the Real ID retooling of state DMV systems may not adequately consider justice and public safety's needs or perpetuate limited legacy JPS interfaces.

The next step is to provide CANDLE capabilities at more DMVs. AAMVA began an infrastructure upgrade, which deploy AAMVA "gateways" in a number of states. These gateways provide motor vehicle

agencies with the ability to exchange digital images but could also serve as a platform for CANDLE sharing. In addition, AAMVA is planning for the next generation of the AAMVA Unified Network Interface (UNI) as web services become readily available. The possibility exists for Nlets and AAMVA to incorporate additional capabilities in the gateway and the next generation UNI services to provide for the intrastate and interstate exchange of standards-compliant CANDLE information on a much broader scale. The resulting real-time data sharing would be streamlined, far more efficient, could decrease system costs considerably and will significantly improve officer and public safety.

II. EXECUTIVE SUMMARY

The CANDLE Project clearly has met the intended goals. Three key goals from the grant proposal are listed below verbatim.

The CANDLE grant application identified the serious problem that "... *information sent outside of a state may be difficult to interpret, wasting time and jeopardizing officer safety.*" The solution was to "*provide driver transactions to NLETS users in standard format*". An important constraint was to do the job right by "*comply[ing] with and contribut[ing] to the current Justice Data Dictionary*".

The CANDLE Project has far exceeded the grant application goals and expectations in a number of measurable ways – in the number of state involved, in the quantity of service transactions implemented, and even in the reach of services by including both interstate and intrastate services.

The initial grant identified the prospect of two pilot states. The CANDLE Project ultimately provided funding and resources for five states. Wisconsin, Delaware, Iowa and Maine have all implemented pilot CANDLE services. In addition, New York State has implemented a fully operational CANDLE interface for all interstate and in-state DMV transactions. New York State also provided substantial state funding to accomplish full implementation (much more than matching funds).

The initial grant identified the need to provide CANDLE capabilities for select driver transactions. The actual CANDLE implementation provides for the complete set of driver and vehicle transactions. New York is even performing state specific updates, e.g. stolen vehicle notification to DMV.

The initial CANDLE grant was intended only for interstate exchange of public safety DMV information. The actual implementation in New York identified the value of using CANDLE XML specifications for both interstate and intrastate exchange. Intrastate transactions did not require data standardization but many of the other transformation and display benefits of XML were realized at essentially no additional cost. One of

the most important findings of the CANDLE Project is that *the best practice for implementing CANDLE is to transform data into CANDLE format immediately upon receipt from DMV or get CANDLE formatted data directly from DMV.*

One of the real success stories of CANDLE was the consensus process that was used. The CANDLE Project Team coordinated extensive conference calls with numerous state public safety representatives to document law enforcement DMV response needs. Two types of driver responses, driver status and driver history, are provided. The status response needs to be as concise as possible because it is used extensively by officers in the field. The CANDLE Project participants worked together to provide the appropriate balance between the need for a short status response and more detailed history response. The conference calls were chaired by an AAMVA subject matter expert that was very familiar with the standardized AMVAA driver transactions. Several states were involved (FL, IA, ME, IL, NY, WI, NV, TN, UT, PA, AZ, GA, VA, MD, ID, IN). Unfortunately, there was limited involvement of state DOT/DMV personnel. The DOT/DMV personnel are typically very familiar with the limitations of state DMV systems and plans and resources for change.

The results of the extensive CANDLE conference calls were codified in the Nlets User and Technical Guide. In addition, Nlets developed GJXDM schemas for the associated transactions which clearly defined the implementation requirements. Nlets coordinated all of the written CANDLE sharing activity using a collaboration server. Nlets also provided CANDLE workshops and presentations at the Nlets annual membership conference and also at the Nlets Technical Conference. The Nlets Technical Conference has become a recognized event of serious public safety system developers from both government and industry.

The proposed technical approach for the CANDLE Project was to create a single gateway to AAMVA standardized driver information through Nlets. As the project began, several problems arose. First, there was not enough information available through AAMVA interface to meet public safety needs. Second, the

state DPS typically wants to control interstate release of information for public safety. Third, the implementation of a central Nlets access point was not conducive to in-state use of the information.

An alternative approach was identified and implemented. The alternative approach was to define XML specifications for DPS implementation. This approach was far more intrusive and required state specific implementation. There was also the need for considerable investment to adopt new technology and standards, e.g. GJXDM. The benefit was comprehensive, standardized DMV data for public safety.

The CANDLE Project is making the national standard driver status and history as highly visible as the national criminal history format. States generally are implementing in-state presentation format to avoid retraining but are using the national standard format for all out of state requests. The national standard view will be very familiar to later adopters because all interstate CANDLE messages are being converted to the national standard view for delivery as text. As a result, states are being advised to have the capability to render in either view for interstate collaboration and to provide the desired display format when rendering XML log file entries.

The CANDLE Project participants have identified several best practices and "lessons learned" (listed below).

- Importance of DOT/DMV involvement
- Greater analysis and specification of AAMVA "codes"
- DMV data changes no longer just text
- More formal change control procedures
- Versioning policies – need to be driven by functional needs in addition to GJXDM versions
- Potential need for CANDLE "certification" process and transaction validation
- New role and content for Nlets "HELP" file

One of the greatest lessons is the importance of greater involvement of the DOT/DMV. DOT/DMV staff can provide information about available data but, more importantly, they are aware of limitations of the DMV systems and planned changes. The next iteration of the CANDLE specifications must include DOT/DMV personnel.

The law enforcement DMV data has different fields that needed to be reconciled between the states. In addition, there are different coded values for many of these fields. For example, the conviction field in a driver history response might have the DUI in one state and the number 5 in another state. These differences were even greater than initially perceived. Future efforts need to consider greater analysis and specification of the codes and close alignment where possible with standardized AAMVA codes. In some cases, the AAMVA codes did not provide the degree of detail required for law enforcement.

While the use of XML provides much greater opportunities to manipulate the resulting data, it also requires coordination for changes. DMV data changes are no longer just text. If changes are made to the data, it may have downstream implications. As a result, changes will have to be more closely coordinated. As the name implies, XML is very extensible. New data can typically be added without adverse impact.

CANDLE continues to be extended on a regular basis. Improvements are managed informally through the Nlets collaboration server. Recommendations are arbitrated by the Nlets staff and periodic conference calls. A more formal Change Control Board is probably needed. Increasingly, changes will be requested which are not backward compatible. Multiple versions will need to be managed and maintained and the Nlets organization has identified the need to address the implications of these decisions more systematically.

Early CANDLE versions were aligned with GJXDM versions. Versioning was driven by technical changes rather than functional needs. Future CANDLE revisions should to be driven by functional needs more than GJXDM versions.

Neither Nlets nor the FBI CJIS Division has a formal process for validating new implementations. This is not surprising given that both the community of fifty states and the technical services have been fixed for many years. New technology is driving many states to dramatically change their implementations of law enforcement DMV services. The implementation of the Real ID Act will compound the pace of these changes. A CANDLE "certification" process would be helpful in this environment to insure complete and accurate operation of the new CANDLE services. The certification process would likely involve the validation of key transactions using XML schema. XML schemas are available for every CANDLE transaction.

As Nlets transactions are standardized through projects like CANDLE, there will be a new role and content for Nlets "HELP" file. There will no longer need to be state specific HELP files but instead there will be a single HELP file entry detailing the national standard presentation format for each transaction.

CANDLE is growing and changing. There is a need to review the early work done under CANDLE, particularly in light of major looming DMV initiatives such as Real ID. Nlets anticipates the need for a future CANDLE Summit that would reach out to a broad group of Nlets representatives and DOT/DMV representatives. The CANDLE Summit would include a broader array of subject matter experts including lead training staff. The CANDLE Summit would include a comprehensive review of CANDLE data content (fields, values). The Summit would also provide a critical review of the national standard presentation format. The end result would be any recommended presentation changes and the associated publicly available XML style sheets.

The fastest and most effective way to extend CANDLE nationwide is to align with the common services offered today through AAMVA. AAMVA's Unified Network Interface already provides many of the standardized DMV services that are envisioned for CANDLE. Alignment with the AAMVA services would create an economy of scale that would benefit all organizations.

There is a great opportunity for a formal CANDLE case study that would show the clear return on investment. New York State has been a great success story and would be an ideal case study. The CANDLE Project has the benefits of both cost saving efficiency as well as the intangible benefits of officer safety improvement. The website, it.ojp.gov, would be the ideal location to post the CANDLE Case Study.

Overall the CANDLE Project has several significant implications for every Nlets member. Each member has the opportunity to use the CANDLE national standard presentation format. There is great potential for synergy with national rap sheet training. Nlets plans to survey the membership to determine the individual state CANDLE implementation plans.

The Real ID Act will force major retooling of DMV systems. The focus of the Real ID will be on the AAMVA interface rather than legacy law enforcement DMV interface. To solve this problem, Nlets members need to support development of a common interface for AAMVA and law enforcement data exchange. In addition, alignment with AAMVA will allow the CANDLE Project to move from five to fifty states as quickly as possible. This strategy requires the adoption of CANDLE standards and specifications by the DMV and a much closer alignment with AAMVA. In addition, there is the need for adoption by leading Nlets "switch" providers. AAMVA has been successful accomplishing national goals with vendor partners by using a "shared" contract approach. The AAMVA model would potentially provide for the one time cost for all states supported by a switch provider.

Nlets recommends the adoption of CANDLE data formats for intrastate use in addition to interstate. Further, Nlets is working to provide CANDLE compliant services at the state DMV. Nlets has identified a technical approach that would leverage the deployment of gateway servers as part of the AAMVA Digital Image Exchange Project to provide a ready CANDLE capability in those states that use a gateway. This would require that enhance AAMVA Image Application to provide CANDLE web services capability, support a direct AAMVA UNI-to-UNI interface to allow the connection of the Driver License Application UNI directly

to the gateway UNI, and enhance the state DPS system to perform CANDLE XML transformations and provision web services. Extending standards-based driver record information exchange using GJXDM V3.0 will increase the efficiency and effectiveness of law enforcement driver-related inquiries and improve officer and public safety. Aligning the interfaces for public safety and AAMVAnet may permit cost savings to DMVs by eliminating an entire tier of technology.

III. PROJECT DESIGN, DATA AND METHODS

A. Background

Justice and Public Safety agencies receive interstate driver/vehicle information in non-uniform, state-specific formats which poses risk to officers and the public. There have been several highly publicized incidents of innocent victim deaths as a direct result of the release of unsafe drivers by and or courts that did not properly interpret confusing DMV related information.

Under the NIJ-funded CANDLE grant, Nlets, AAMVA and participating states used AAMVA's driver/vehicle information model to successfully convey uniform responses to law enforcement over Nlets in a format compliant with the DOJ Global Justice XML Data Model (GJXDM) information sharing standard. The New York State Police now both receives all NY DMV data and delivers it interstate via Nlets in CANDLE format. Likewise, Delaware, Iowa and Wisconsin have made significant progress toward achieving interstate CANDLE-compliant data exchange.

By way of explanation, motor vehicle departments (DMVs) historically maintained separate system interfaces to support JPS access to driver and vehicle information. The DMV interfaces for law enforcement were developed first and did not require uniform responses across states, as they were developed in an era where dispatchers "interpreted" the data. The AAMVA network was created later and was based upon full automated data exchanges supported by common interface software from AAMVA.

It is possible to provide for the exchange of standards based data directly from the state DMV and ultimately to provide common shared services for DMV data to include JPS. A related goal is that adoption and deployment of CANDLE capabilities by state DMVs will permit the broadest and most rapid deployment of these capabilities. Conversely, the failure to coordinate with major DMV initiatives, particularly the Real ID Act, will almost certainly result in JPS' needs being less than fully considered as the state DMVs

modernize. Nlets will work with AAMVA to insure that these needs are fully considered as AAMVA defines the Real ID deployment strategy for interstate DMV information sharing.

Law enforcement agencies that have adopted CANDLE standards have generally determined that CANDLE benefits are just as applicable to intrastate transactions. It is possible to extend the exchange of driver/vehicle information in a common format between public safety agency groups with strong shared interests, and to realize greater efficiency and economy by eliminating the layer of technology presently used to exclusively support law enforcement access to driver and vehicle information.

AAMVA is currently completing the upgrade of the AAMVA network infrastructure under the Network Control System II (NCS II) project. As this project is completed, AAMVA will be better positioned to introduce next generation technology for DMV exchanges, particularly XML web services. In addition, AAMVA has several projects underway or planned that will result in changes and improvements to AAMVA capabilities. The most immediate is the Digital Image Exchange Project. The Digital Image Exchange Project provides limited funding for all 50 states to implement driver photo exchanges between DMVs for the purpose of licensure fraud prevention. Under this project, AAMVA is providing to each state, the option of the deployment of a "gateway" system that interacts with the DMV driver applications and image systems to exchange images. The gateway system can serve as a platform for providing access to DMV applications which makes it ideal for implementation of CANDLE capabilities.

With the implementation of NCS II, AAMVA will have the infrastructure in place to begin the development of the next generation of the Unified Network Interface (UNI). With the advent of web services, significant UNI improvements are possible and the potential exists to incorporate CANDLE capabilities in the improvement process.

B. Project Design

The Initial CANDLE project design was to provide a central gateway which would provide for the translation of AAMVA data into Nlets XML formats. While this approach was technically sound, it did not consider the political reality that DMV data is highly controlled and is typically released to public safety under close regulation.

During the initial phase of the CANDLE project, two alternative implementation designs emerged. New York decided to format data in the CANDLE format directly at the DMV. Thus, the NY State Police received all DMV data, both intra and interstate, in CANDLE format. The CANDLE formatted data was transformed for in-state use to a text format similar to the existing in-state format to minimize retraining, while interstate Nlets requests were transmitted in the CANDLE format from the DMV.

The other CANDLE implementation approach was to accept DMV data in a current legacy format and transform it into CANDLE format at the Department of Public Safety (DPS). This approach had several limitations. First, the law enforcement data provided by the DMV was generally not as complete or uniform as specified for CANDLE because the state law enforcement interface predated the standardized AAMVA data formats. Second, a substantial amount of effort was required to transform the data and translate field codes. Third, any potential benefits to the DMV were eliminated by integrating the CANDLE services into a DPS system.

To achieve the full benefits of CANDLE formatting, CANDLE-compliant data exchanges should originate from the DMV. Since CANDLE data is accessible via the AAMVA interface and aligned with the AAMVA format, the simplest way to offer a single, pervasive solution is to adapt the AAMVA interface to retrieve the necessary DMV data.

Nlets and AAMVA will have established the following objectives:

- Develop a CANDLE capability through the state DMV gateway and/or next generation AAAMVA Unified Network Interface (UNI)
- Implement both interstate and intrastate CANDLE services
- Coordinate the inclusion of CANDLE capabilities in the AAMVA modernization plans associated with the Real ID Act

Nlets has incorporated the standards, guidelines, and best practices established through the Global Justice Information Sharing Initiative including compliance with the GJXDM standard and use of a Service Oriented Architecture approach to maximize information sharing efficiency. Both Nlets and AAMVA have been strong and active supporters of the Global Advisory Committee. Both organizations have been founding members of the Global Justice XML Data Model initiative and related OJP initiatives. Nlets and AAMVA will continue to fully incorporate the other "building block" initiatives that are being sponsored by OJP. Nlets is also working with Interpol to both influence and incorporate international public safety standards. Nlets has been very aggressive in adopting a Service Oriented Architecture (SOA) and supports the largest national level deployment of GJXDM based exchanges with nearly a million XML rap sheets alone being transformed and exchanged each month. Nlets has also been very aggressive in adopting industry standards such as web services. Nlets has contributed significantly to the efforts of the Messaging Focus Group which is defining the next generation service interaction profiles for the Justice Reference Architecture.

C. Project Data

Three views of CANDLE data presentation have emerged.

- Represent interstate and intrastate data using the "in-state" format
- Represent interstate and intrastate data using the national standard format (Nlets text style)
- Represent intrastate data using "in-state" format and interstate data using the national format

Most states have been adopting the first view. No doubt this is an initial reaction to the need to avoid re-training. Over time the use of the national standard format with state extensions will be increasingly common. The use of separate in-state and interstate formats may also be widely adopted because states seem to like to look at residents from a different perspective anyway. The same issues exist with the rap sheet. In the case of the rap sheet , states are quickly moving to the national format. Samples of driver and vehicle responses using the Nlets CANDLE national standard format are provided in figures 2 and 3. A sample of the vehicle response in the Nlets text style is provided in Figure 1.

```
RR.NYDMVNY00
01:15 04/27/05 00750
01:15 04/27/05 00168 AZNLETS20
TXT
NYMV RVEH LETS 0415
 NPLDMV510 16
HEDR D09031 12865 343173-76
LIC/DMV510. LIY/013003. LIT/PC.
DIAL,TESTD
SWAN STREET. ALBANY NY. 12228
DOB/050176. SEX/M
VIN/DIALTEST10. VYR/94. VMA/SATU. VST/2D. VCO/RED
INS/400  ACE FIRE UNDERWRITERS INS
STATUS/:EXPIRED
```

Figure 1 Non-CANDLE Registration Response Sample

*** REGISTRATION DETAILS ***
REGISTRATION ID: S014163392485654333
VEHICLE REGISTRATION EFFECTIVE DATE: 2005-11-15
VEHICLE REGISTRATION EXPIRATION DATE: 2006-07-19
REGISTRATION JURISDICTION CODE (LIS): NY
REGISTRATION STATUS: VALID

REGISTERED TO: JOSEPH TEST
DATE OF BIRTH: 1982-10-12
GENDER: MALE
MAILING ADDRESS: 444 HOLLAND AVE 2; LACKAWANNA,NY 14218
VEHICLE REGISTRATION PLATE ID: ABC123
VEHICLE REGISTRATION PLATE TYPE: PASSENGER

*** VEHICLE DETAILS ***
VEHICLE ID: 1GNDU03E7WD240333
VEHICLE VINA: CHEV-VEN
VEHICLE MAKE CODE: CHEV
VEHICLE MAKE TEXT: CHEVROLET
VEHICLE MODEL CODE: VEN
VEHICLE MODEL YEAR: 1998
VEHICLE MODEL TEXT: VENTURE
VEHICLE STYLE CODE: SW
VEHICLE PRIMARY COLOR: GREEN

*** INSURANCE DETAILS ***
INSURANCE CARRIER: FARMINGTON CASUALTY CO

Figure 2 CANDLE Vehicle Registration Response Sample

NAME: MARVIN K TEST
PRIMARY CONTACT ADDRESS: 2070 CLINTON AVENUE; BRONX,NY 10457
COUNTY: BRONX
LOCATION COUNTY: BRONX
DATE OF BIRTH: 1974-07-18
OTHER ID: 711671383
HEIGHT: 507
EYE COLOR: BROWN
GENDER: MALE

*** DRIVER LICENSE DETAILS ***
DRIVER AUTHORIZATION ID: H017108376373477874
ISSUE DATE: 2004-03-03
EXPIRATION DATE: 2012-07-18
DRIVER ENDORSEMENT: PASSENGER
DRIVING RESTRICTION CODE: NOT VALID FOR AIR BRAKES
DRIVING RESTRICTION CODE: NO VEHICLE OVER 18,000 LBS
DRIVING RESTRICTION CODE: ADULT SEATING CAPACITY - 7 OR LESS
DRIVER LICENSE PERMIT QUANTITY: 0
DRIVER LICENSE COMMERCIAL CLASS CODE: C
DRIVER LICENSE NON-COMMERCIAL CLASS: C
DRIVER LICENSE NON-COMMERCIAL CLASS CODE: CDL *C*
DRIVER LICENSE COMMERCIAL STATUS: REVOKED
DRIVER LICENSE NON-COMMERCIAL STATUS: VALID
DRIVER HISTORY CONVICTION QUANTITY: 2

DRIVER HISTORY ACCIDENT QUANTITY: 0
DRIVER HISTORY WITHDRAWAL QUANTITY: 1

*** DRIVER CONVICTION ***
CONVICTION DATE: 2005-10-24
CONVICTION LOCATOR REFERENCE ID: 5102404001
ORGANIZATION TYPE: BRONX COUNTY, ADMINISTRATIVE ADJUDICATION
OFFENSE DATE: 2005-10-21
DRIVER CONVICTION TEXT: LEAVING THE SCENE OF A PROPERTY DAMAGE INCIDENT WITHOUT REPORTING

*** DRIVER CONVICTION ***
CONVICTION DATE: 2005-02-02
CONVICTION LOCATOR REFERENCE ID: 5020204003
ORGANIZATION TYPE: BRONX COUNTY, ADMINISTRATIVE ADJUDICATION
OFFENSE DATE: 2004-05-23
DRIVER CONVICTION TEXT: OPERATION OF A MOTOR VEHICLE WHILE USING A HAND HELD MOB ILE PHONE

*** DRIVER LICENSE WITHDRAWAL ***
WITHDRAWAL EFFECTIVE DATE: 2005-06-07
WITHDRAWAL REINSTATEMENT DATE: 2005-07-22
WITHDRAWAL REASON REFERENCE CODE: FAILURE TO PAY FINE (SCOFFLAW)
WITHDRAWAL ACTION: SUSPENSION
WITHDRAWAL LOCATOR REFERENCE: K732370

Figure 3 CANDLE Driver History Response Sample

D. Project Methods

In parallel with the deployment of CANDLE capabilities, Nlets intends to work with AAMVA to incorporate JPS requirements into the Real ID implementation planning process. This will involve a broad outreach effort within the DMV community as well as a revisiting of the CANDLE specifications with a broader Nlets constituency and inclusion of DMV subject matter experts. The CANDLE specifications may be extended to include data that is typically used only used by in state law enforcement such as handicap placard information.

The CANDLE specifications were incorporated into the Nlets User and Technical Guide and the entire Nlets community is familiar with that document. Since that time, the justice community has adopted the Information Exchange Package Documentation (IEPD) methodology and guidelines. Formal Information

Exchange Package Documentation (IEPD) will need to be developed as part of the next generation of CANDLE definitions.

The CANDLE Gateway approach assumes that DMV data is accessed through the AAMVA Unified Network Interface. Currently, UNI does not provide access to all of the required information, particularly about vehicles. The UNI data limitations need to be identified and a future mitigation strategy developed.

Once DMV data is accessible, the second problem is finding a common mechanism to transform DMV data into CANDLE format. Three major components are associated with implementation of CANDLE II gateway capabilities – the AAMVA Unified Network Interface, the AAMVA Gateway system and the DPS CANDLE-format processes. These three components can be managed independently.

The AAMVA UNI component must be modified to provide to directly exchange AAMVA formatted data with the CANDLE Gateway system. This effort requires the direct involvement of AAMVA as a critical CANDLE partner, in order to make all required changes to the proprietary UNI software.

AAMVA is providing its Gateway system to DMVs under Department of Transportation sponsorship. The DMV-owned/operated AAMVA Gateway platform allows digital image exchange between DMV offices. CANDLE Gateway services would also employ the same UNI interface. The CANDLE Gateway would bridge between the AAMVA UNI interface and a DPS web services interface. The CANDLE Gateway would transform the AAMVA formatted data into CANDLE XML.

Finally, CANDLE states or their technology partners will implement CANDLE DPS processes. Interstate exchange of CANDLE information over Nlets would thereafter be very straightforward since the DPS would already receive DMV data in CANDLE format. In-state processing includes application of a style sheet to transform data into the desired state format. The in-state style sheet could also be applied to CANDLE formatted data received from another state via Nlets (Figure 4 Future CANDLE Architecture).

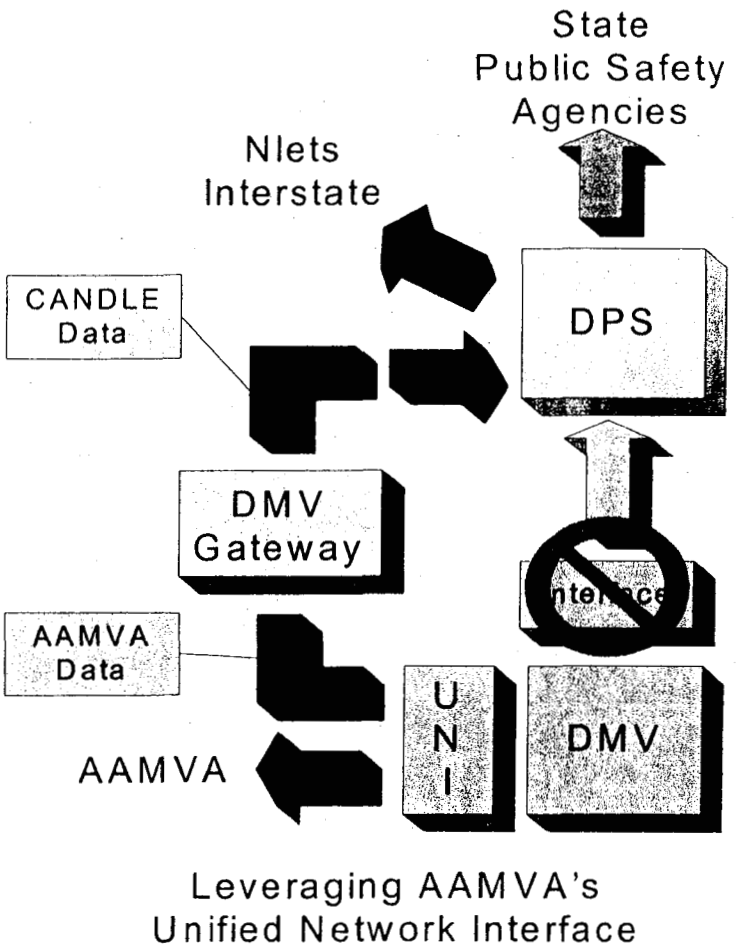


Figure 4 Future CANDLE Architecture

The Unified Network Interface (UNI) provides a call level interface that resides directly on the DMV computing platform. UNI exchanges transactions through the AAMVA Network Control System II (NCS II). At present, a UNI-based system may communicate to another UNI-based system only through AAMVA's NCS II. As it would be inefficient to use the intermediate NCS system for in-state exchanges, it will be necessary to enhance UNI to provide a direct UNI-to-UNI capability allowing access from the Gateway system in a manner sufficient to support the high volume public safety requests for CANDLE data.

CANDLE data formats have been aligned with the AAMVA Message Interchange Envelope format (AMIE) and Messages over AAMVAnet using XML (MAX), which makes it possible to develop an additional

CANDLE gateway application that both transforms CANDLE requests into the UNI call list format and conversely, transforms the UNI call list response data to the CANDLE format. With the CANDLE gateway application in place, state DPS systems could be modified to exchange CANDLE data between Nlets and the DMV CANDLE gateway.

In-state DMV requests need to be converted from legacy format to the CANDLE request format for in-state enforcement agencies to use this capability, Likewise, CANDLE responses need to be converted to text for display on in-state legacy terminals. CANDLE request creation is straightforward because of its small size and the precise format of legacy requests. Response transformations can be performed simply because of the ease with which a style sheet can be applied to the CANDLE response.

Provided the national CANDLE text format is acceptable to the in-state agency considering such use, existing CANDLE style sheets can be applied. If the current State format is desired, a custom style sheet can be created matching the legacy text format. As an additional service, the AAMVA Gateway could also be used to provide access to in-state DMV images for local public safety use, which could be easily accomplished by extending CANDLE specifications to support image exchange. At present, the DMV image exchange agreement prohibits interstate exchange of DMV photos except for license fraud investigation. The ongoing national discussions on identification management, security, and privacy issues eventually should result in regulation changes enabling interstate photograph access and exchange.

Web services will provide the exchange mechanism between the CANDLE gateway and state DPS system. Each interested State DPS system would need to support web services. The AAMVA Gateway is a Windows server with development tools to implement XML transformations and web services easily. The newly-installed AAMVA gateway systems will already contain the latest UNI software (version 4.0) needed to support the AAMVA Image Application.

IV. PRESENTATION OF FINDINGS

A. Findings

It is possible to develop a gateway system that can be used to broadly deliver CANDLE web services capabilities. To meet required performance, AAMVA UNI modifications would be needed to support a direct UNI-to-UNI interface; this would allow the Driver License Application UNI to connect directly to the gateway UNI. State DPS system enhancements to perform CANDLE XML transformations and provision of web services will also be required. Taken altogether, these enhancements would provide standard CANDLE transactions and data to JPS across the United States. Further, if these services meet all JPS needs, it may be possible to eliminate the legacy State DPS DMV law enforcement interfaces and thereby provide considerable long term savings. Most important, making the standard CANDLE XML format generally available through this proposal will provide uniform DMV responses to JPS improving efficiency and enhancing the public and the line officer's safety.

It is possible that current AAMVA applications may not provide all of the data required by existing in-state JPS applications, e.g., handicap placard status. Alternatively, if *all* law enforcement services can eventually be provided through the CANDLE gateway, DMVs may be able to eliminate separate law enforcement/State DPS interfaces and need only employ the UNI interface to deliver both AAMVA and JPS services. This innovation would eliminate DMV costs to maintain the legacy state DPS interfaces and constitute a significant long term cost saving benefit.

B. Dissemination Strategy

Nlets and AAMVA provided periodic updates on the CANDLE Project to their respective memberships and stakeholder organizations through internal communication mechanisms and web sites. Both organizations provided the oversight and approval of their respective governing bodies.

Nlets was highly effective in communicating information about the CANDLE project at both the Nlets Annual Conferences and the recent Nlets Implementers Conference. The Nlets Implementers Conference is second only to the GJXDM User Conference in terms of reaching implementation managers and technical staff with directly appropriate responsibilities. For the past two years, approximately 150 people have attended the Nlets Implementers Conference in January and heard both technical and managerial presentations concerning CANDLE. Nlets also has worked directly with leading vendors in an effort to seed CANDLE capabilities into the vendor products to be incorporated as part of the vendor product release.

Nlets has provided CANDLE presentations in a number of national conferences as an example of a success story and to communicate best practices.

Nlets state members have in turned communicated the CANDLE message within the states. The most significant example is the New York State Police (NYSP). NYSP received a small secondary CANDLE grant from Nlets which it matched with funds and internal resources. The result is a CANDLE compliant capability being provided by the NY DMV and providing all state law enforcement DMV access. NYSP has effectively provided a ten fold match and deployed GJXDM compliant services for all interstate and intrastate DMV law enforcement services.

At the request of Nlets representatives, several state public safety agencies are working directly with the DMV counterparts to implement CANDLE capabilities. Nlets has tremendous access and influence with state representatives to achieve broad dissemination.

C. Performance

The table below summarizes the findings of the CANDLE grant in terms of key performance criteria.

Performance Criteria	CANDLE Value
A. Understanding of the problem and its importance	

Understanding of the problem and its importance	The problems of confusing interstate responses are a daily operational concern that has been repeatedly expressed by Nlets on behalf of the field officers.

B. Quality and Technical Merit

Soundness of methodology and analytic and technical approach.	The CANDLE Project developed a sound methodology that was extensively vetted within the AAMVA and Nlets organization.
Innovation and creativity	Nlets leads the public safety community in developing innovative information sharing solutions based on GJXDM standards. The CANDLE Project demonstrated creative thinking in integrating two sets of needs (DMV and public safety) into a common solution.
Feasibility of proposed project and awareness of pitfalls.	AAMVA and Nlets developed a technical approach avoided the numerous technical and organizational pitfalls.
Awareness of the state of current research or technology.	CANDLE leveraged industry research and development in GJDXM and the NIJ AISLE Project. The CANDLE participants are leaders in public safety and participate in standards committees and working groups.

C. Impact of Project

Potential for significant advances in the field.	As it is more broadly adopted, CANDLE will dramatically improve the effectiveness of public safety information sharing and improve officer safety. Driver and other motor vehicle related transactions account for over half of the Nlets daily volume.
Potential for significant advances in scientific or technical understanding of the problem.	CANDLE gives key state IT professionals in both DMV and public safety agencies a greater understanding of technical issues and the opportunity to develop corresponding best practices. The promulgation of XML to public safety organizations will promote information sharing through state integrated justice initiatives.
Relevance for improving the policy and practice of criminal justice and related agencies and improving public safety, security, and quality of life.	CANDLE addresses the need for public safety agencies to improve information sharing and leverage the benefits of Internet driven technologies. Likewise, members have expressed the benefits of providing consolidated services for both the DMV and public safety information needs.

Relative importance of the criminal justice or public safety problem issues to be addressed by the proposed effort.	CANDLE addresses a mainstream public safety problem, effectively and efficiently obtaining driver information, and fosters information sharing to improve homeland security. CANDLE also responds to the need to lower costs, increase capabilities, and build technology standards
Affordability and cost-effectiveness of proposed end products, when applicable (e.g., purchase price and maintenance costs for a new technology or cost of training to use the technology).	CANDLE employs standards and technology that improve service and lower cost. The replacement of proprietary standards with open standards will significantly lower costs by providing greater competition and standardization. Common AAMVA/Nlets interface will lead to long term reductions in maintenance costs.
Perceived potential for commercialization and/or implementation of a new technology (when applicable).	Nlets provides solutions as "open source" to public safety agencies. In addition, Nlets is contracting with key vendor partners to insure that states can obtain CANDLE services as an integrated part of the products that they routinely purchase.

D. Capabilities, Demonstrated Productivity, and Experience

Qualifications and experience of proposed staff.	The participants are leading public safety and DMV IT practitioners.
Demonstrated ability of proposed staff and organization to manage the effort.	The project staff and organizations successfully managed the CANDLE Project in the same manner as a number of leading edge national DMV and public safety technology initiatives.
Adequacy of the plan to manage the project, including how various tasks are subdivided and resources are used.	Nlets developed a detailed task breakdown with associated resources. The experience of the grant management provided high assurance of the adequacy of our management plan. Both AAMVA and Nlets have Technical Committees in place that provided guidance and oversight insuring adequate management.
Successful past performance on NIJ grants and contracts	Nlets very successfully performed the CANDLE grant and the previous AISLE grants sponsored by NIJ and BJA.

E. Budget

Total cost of the project relative to the perceived benefit.	The return on investment for CANDLE is substantial because it "seeds" capabilities that will affect every law enforcement organization in North America. This return is further multiplied by the impact on the

	AAMVA community.
Appropriateness of the budget relative to the level of effort.	Nlets provided downstream resources to numerous states to further make the relative cost very reasonable.
Use of existing resources to conserve costs.	CANDLE built upon the equipment and expertise deployed as part of the NIJ AISLE effort. NLETS committed the NLETS staff and their extensive expertise at no cost.

F. Dissemination Strategy

Well-defined plan for the grant recipient to disseminate results to appropriate audiences, including researchers, practitioners, and policymakers.	Nlets provided direct access to state law enforcement leadership to provide effective dissemination through conferences and numerous communications capabilities.

Performance Objective	
Develop, evaluate, or test technologies and tools that can increase the effectiveness of law enforcement agencies' and officers' use of information to guide decision making to reduce and prevent crime.	
Performance Metric	**CANDLE Value**
Relevance to the needs of the field as measured by whether the grantee's substantive scope did not deviate from the funded proposal or any subsequent agency modifications to the scope.	Nlets has a record of delivering more than initially specified in project scope. The original CANDLE grant identified two states for pilot deployment of driver related transactions. Nlets delivered driver and vehicle capabilities to five states.
Quality of the research as assessed by peer reviewers.	Nlets has consistently delivered the highest quality engineering services. The confidential survey after the Nlets Implementers Conference had only one person out of nearly a hundred that said that they would not return to the conference again.
Quality of management as measured by whether significant interim project milestones were achieved, final deadlines were met, and costs remained within approved limits.	The CANDLE Project was completed within budget. Under the auspices of the CANDLE project and with matching funds, the State of New York converted all DMV law enforcement transactions to a new CANDLE compliant interface in less than a year.

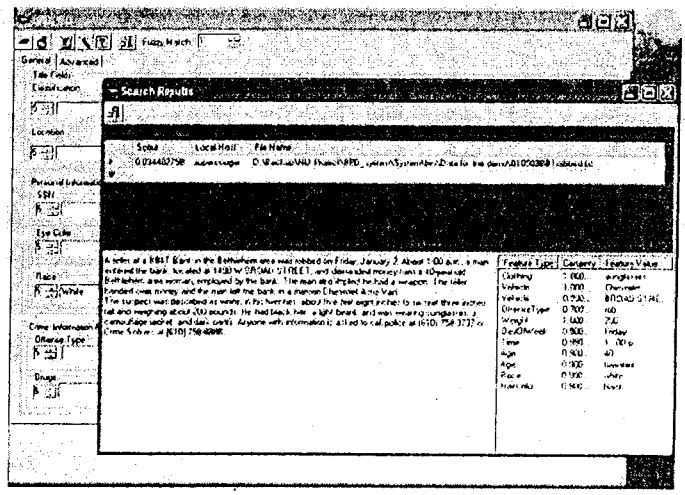
V. IMPLICATIONS FOR POLICY AND PRACTICE

The CANDLE implications for both policy and practice are enormous. When fully deployed, CANDLE will improve the effectiveness of nearly every justice and public safety professional in the nation. It will make these professionals more efficient in this resource poor community while saving lives by avoiding terrible mistakes in decoding infrequently used interstate data. Even states that do not adopt CANDLE are benefiting by the standardized text responses generated by Nlets in response to GJXDM compliant CANDLE states. At present New York is the only state producing all law enforcement DMV responses in the national standard CANDLE format so there is no apparent benefit beyond an easier to read New York DMV record. But other states are close to providing CANDLE responses. As the second and third states produce the same format, it will be readily apparent that there is a national standard that can be adopted and for which officers can be effectively trained. States are seeing a similar situation with the national standard rap sheet and will increasingly expect interstate data to be represented in a common national standard format with GJXDM as the underpinning.

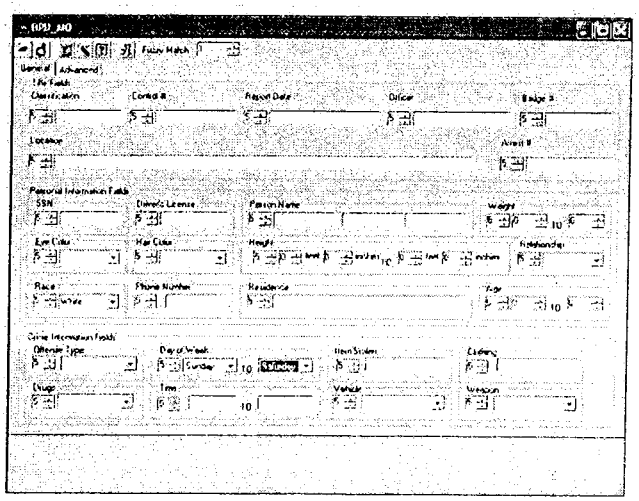
The policy implications associated with the Real ID Act implementations are even greater. State DMVs will be struggling with the major re-engineering efforts that will be needed to support the identity verification requirements of the Real ID Act. This is estimated to be hundreds of millions of dollars. State DMVs will implement funded Real ID requirements first with potentially disastrous consequences for legacy law enforcement interfaces. In many cases these specialized law enforcement DMV interfaces were developed years ago by a staff that may have since retired. Changes are problematic and re-engineering funding may not be available. CANDLE provides a framework and specification for the law enforcement DMV exchange requirements that can be adopted as part of Real ID re-engineering efforts. The goal of the Real ID Act is greater homeland security and public safety. It is important not to lose sight of that goal as it is translated

into changes in DMV systems to improve identity. CANDLE will permit the downstream benefits of Real ID improvements in identity and criminal information to be delivered to the officer on the street where it will be most effective.

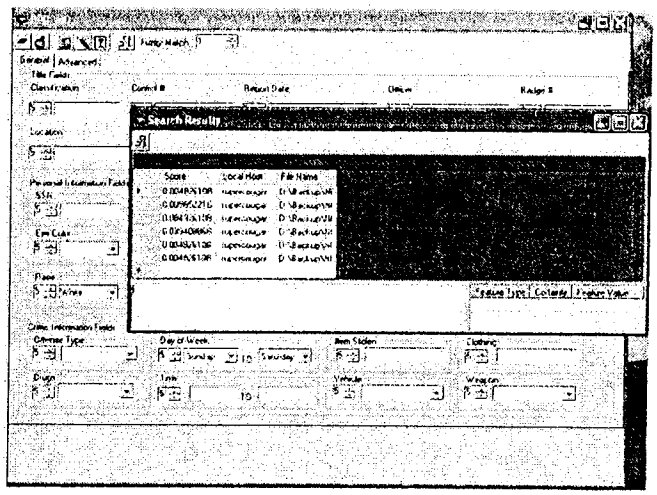
Fuzzy Match - Result



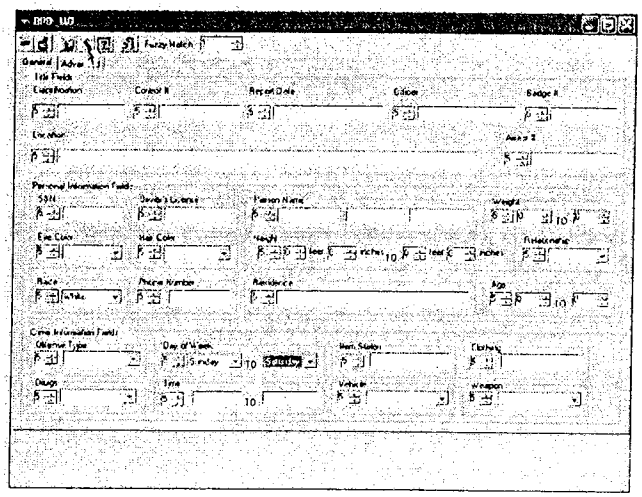
Range Search



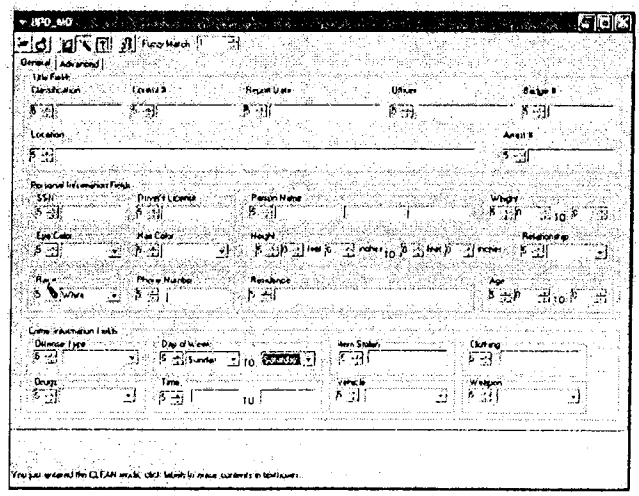
Range Search - Result



Erase Mode

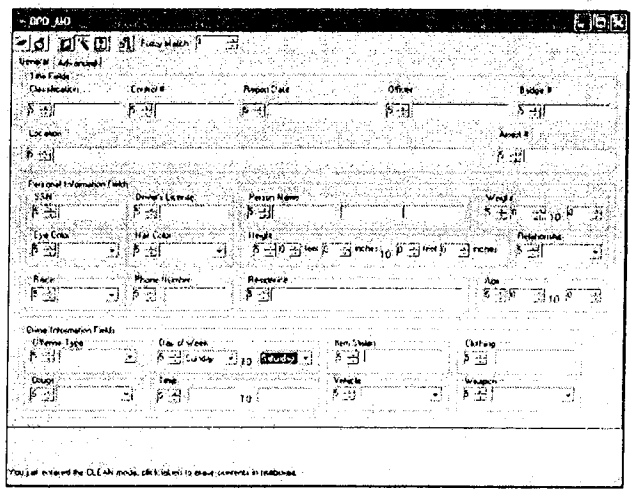


Clear A Single Field



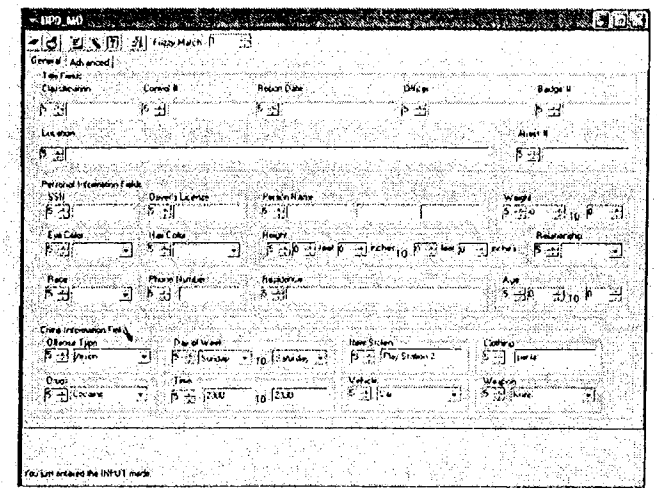
1/25/2005 50

Clear A Single Field - Result

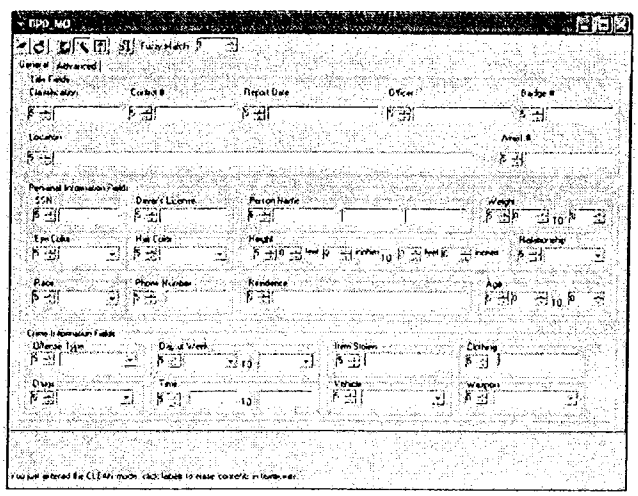


1/25/2005 51

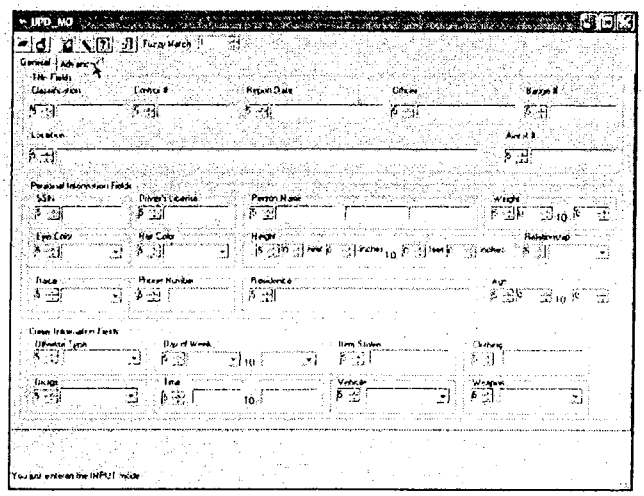
Clear A Field Group



Clear A Field Group - Result

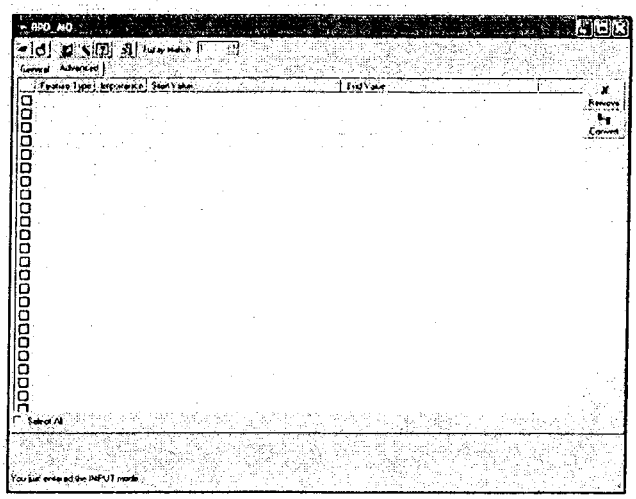


Advanced Search



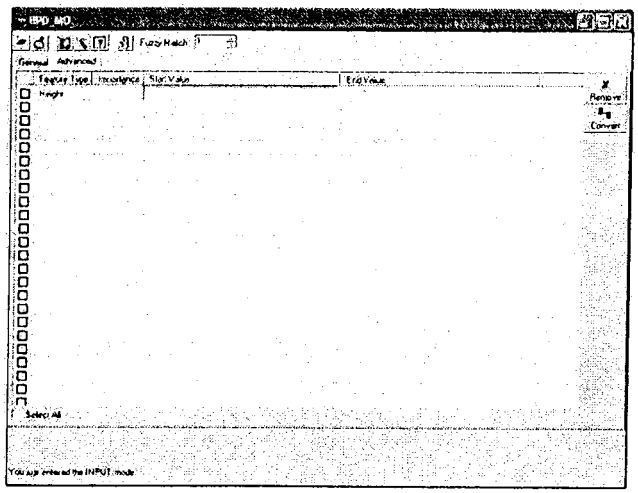
1/25/2005 54

Advanced Search - GUI



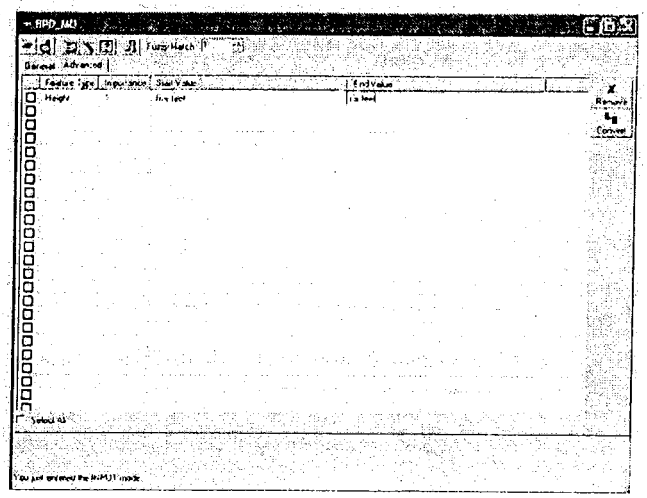
1/25/2005 55

Advanced Search
– Select A Feature Type



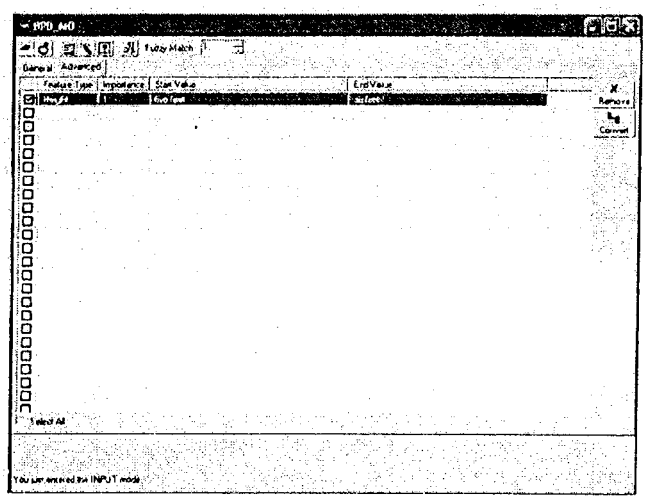
1/25/2005 56

Advanced Search
– Input Value(s)

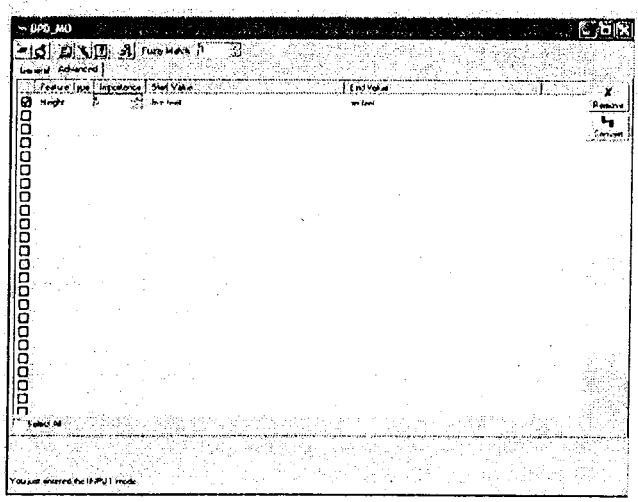


1/25/2005 57

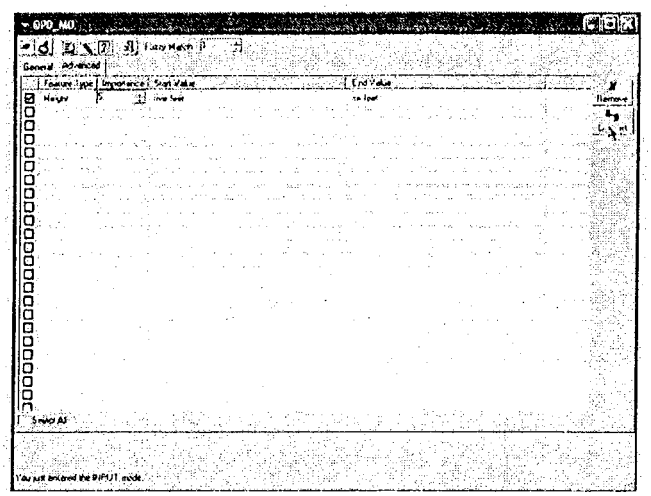
Advanced Search – Checkbox



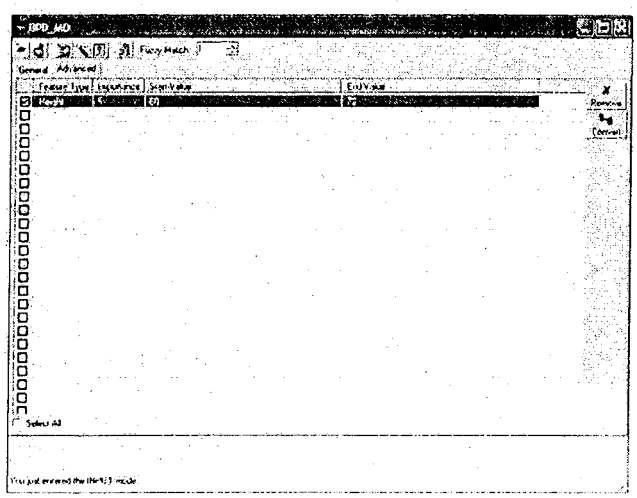
Advanced Search – Change Importance



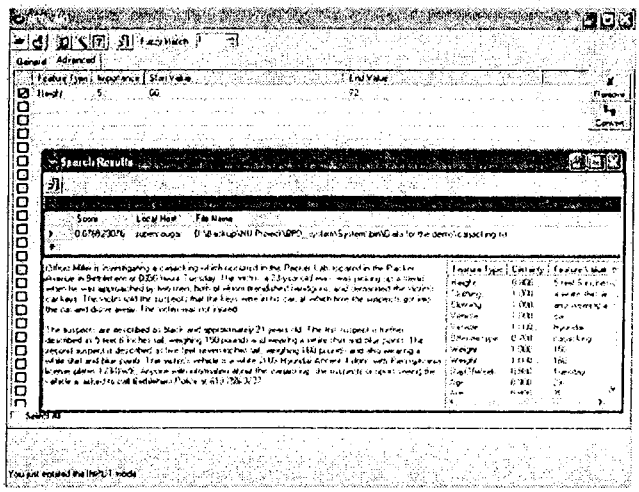
Advanced Search
– Convert Values



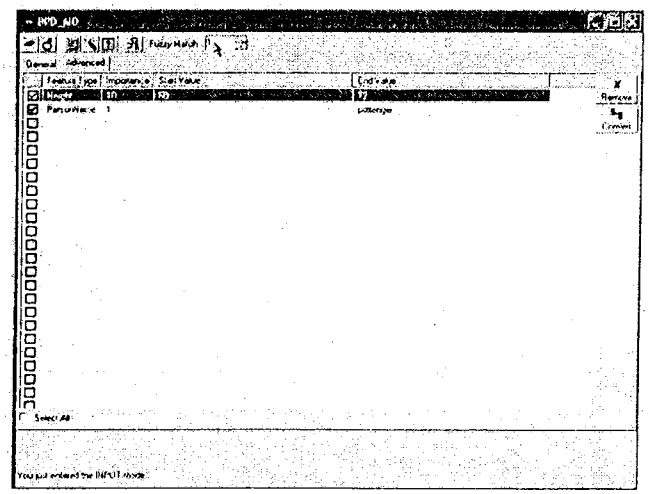
Advanced Search
– Convert Values (Result)



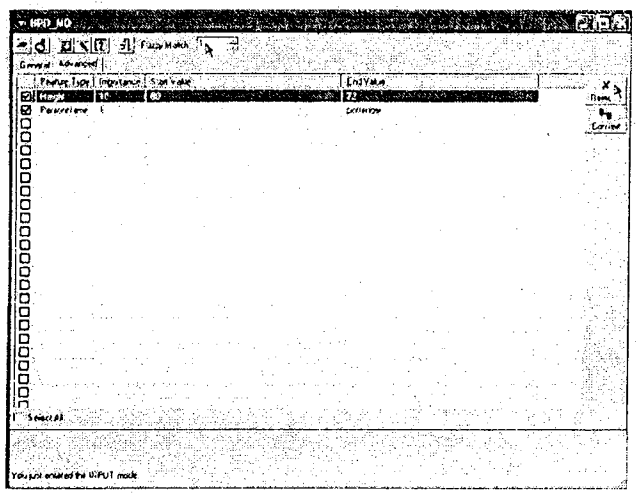
Advanced Search – Search the Feature



Advanced Search – Fuzzy Search

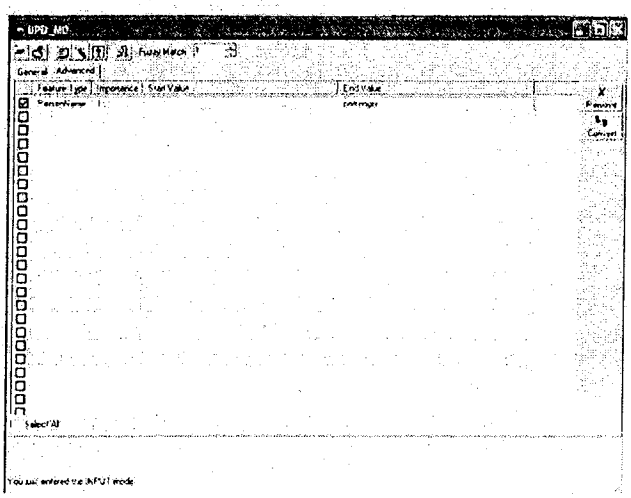


Delete

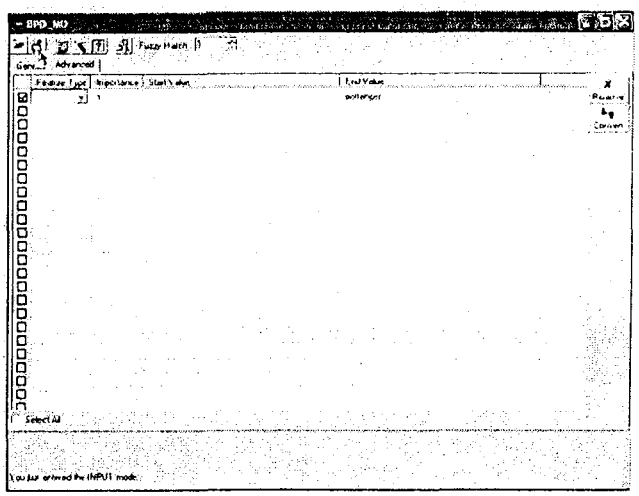


Highlighted features can be deleted one at a time.

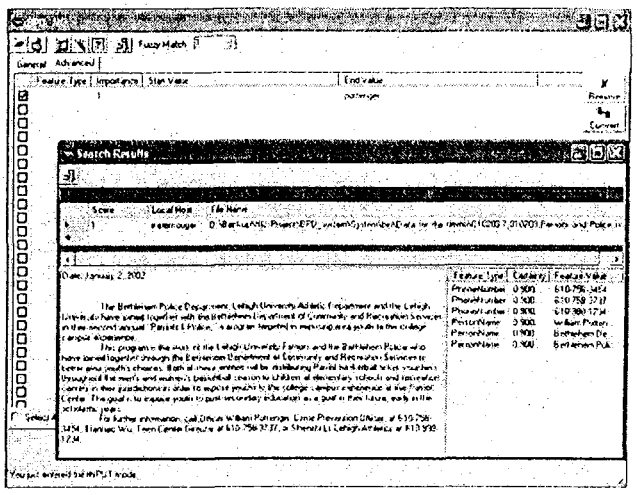
Delete - Result



Advanced Search
– Search Without Feature Type



Advanced Search
– Search Without Feature Type (Result)



Appendix C – Solution File Descriptions

You need both "SupportFiles" and "System" folders and all files in them to compile the source code.

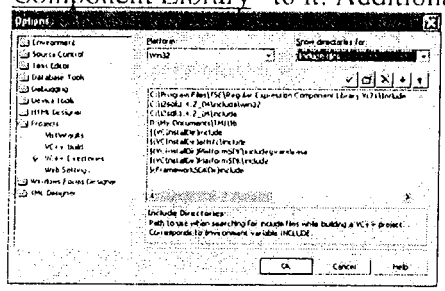
BPD_FE and BPD_MO

Open the solution file

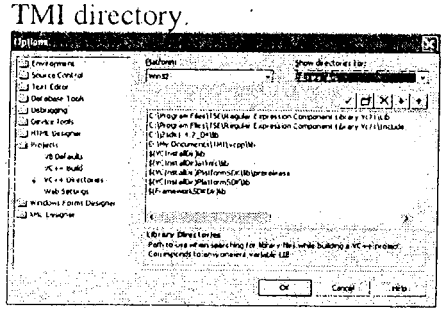
The solution file "BPD_IE.sln" is located in the "BPD_FE" folder. The file can be opened the same either by going to that folder to open the BPD_IE.sln or by running the "openSolution.bat."

Configure the solution file

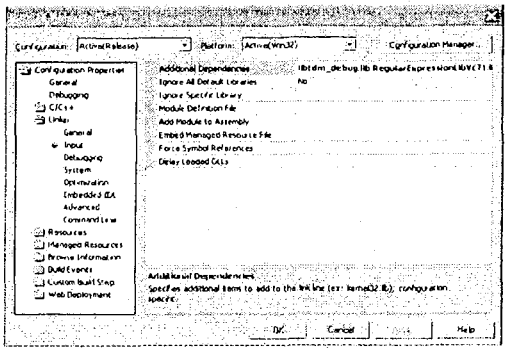
1. From the main menu, go to "Tools"→"Options"→ "Projects" → "VC++ Directories" Under the "Show Directories for" tab:
   - Choose "Include files", and add the "Include" folder of the "Regular Expression Component Library" to it. Additionally, add the "/lib" folder of the TMI directory.



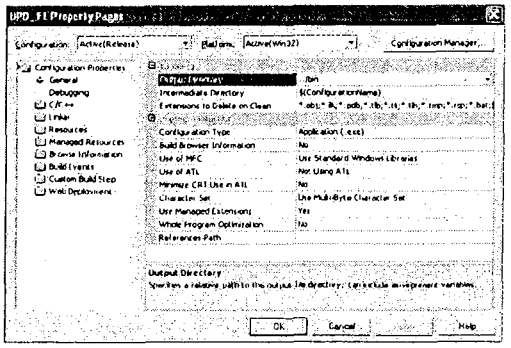
   - Choose "Library files", and add the "lib" and "Include" folders of "Regular Expression Component Library" to it. Additionally, add the "/vcpp/lib"folder of the TMI directory.



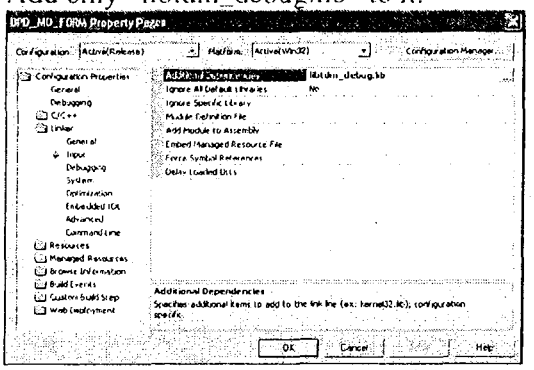
   - Click "OK" to save your configuration.
2. Select the "BPD_FE" project in the solution explorer and then "Project"→ "Properties"→ "Configuration Properties" → "Linker" → "Input" → "Additional Dependencies"
   Add "libtdm_debug.lib" and "RegularExpressionLibVc71.lib" to the list of dependencies.



3. Select the "BPD_FE" project in the solution explorer and then "Project"→ "Properties"→ "Configuration Properties" → "General" → "Output Directory" Add "./bin" to it.



4. Select the "BPD_MO" project in the solution explorer and then "Project"→ "Properties"→ "Configuration Properties" → "Linker" → "Input" → "Additional Dependencies"
Add only "libtdm_debug.lib" to it.



Distribution

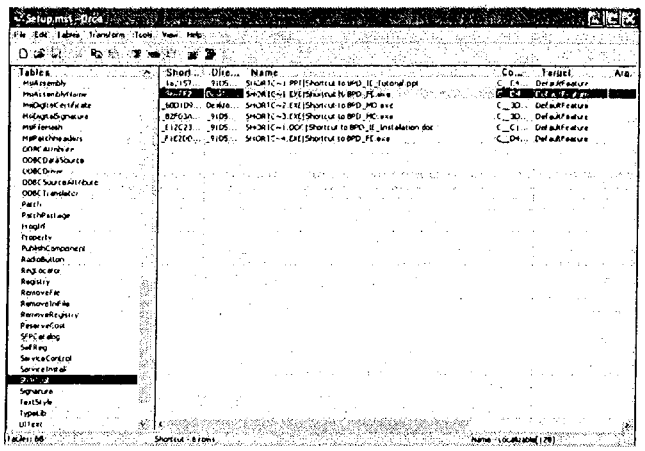
Build the main setup file "setup.msi"

Open "BPD_IE.sln" under the "Setup\Setup" folder. Build the project. After that, a file "setup.msi" will be generated in "Setup\Setup\Release" folder. When the user installs this program, there will be two shortcuts created on the user's desktop. However, the default

properties of these two shortcuts do not support drag and drop functionality. You must use ORCA to modify it manually before you distribute the setup file.

You can download and install ORCA.msi from HERE.

1. Run ORCA
2. Open the "setup.msi" file in the "Setup\Setup\release" folder
3. Find the Shortcut category
4. Find the items with name "Shortcut to BPD_FE.exe", and then change "defaultFeature" to "[TARGETDIR]\bin\BPD_FE.exe" under "Target".
5. Find the items with name "Shortcut to BPD_MO.exe", and then change "defaultFeature" to "[TARGETDIR]\bin\BPD_MO.exe" under "Target".
6. Save the setup.msi



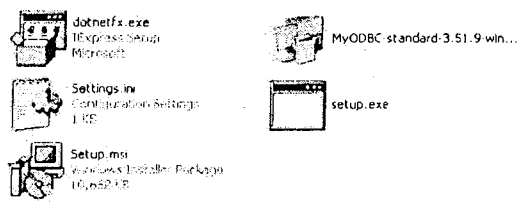
Prepare your own setup wrapper

You can wrap many individual setup programs into a single setup process. After you install one, the next one will automatically start to install. Make sure the user installs all necessary software for your system by using this wrapper.

Open "setup.sln" under the "Setup\Bootstrapper". Modify the code if you would like, and then build the project. A file "setup.exe" will be generated under "Setup\Bootstrapper\Release" folder.

Run "distribute.bat" under "Setup". It will automatically collect all installation files for the BPD_IE systems into the folder "BPDIE_binary_release". Notice that setup.msi is only one of the installation files generated.

This folder and all files located within it is your distribution package. Zip the files or copy them to the computers where you would like to install the BPD_IE client system. The following figure shows all files that should be in "BPDIE_binary_release".



Appendix D – Classes

11.1 *BPD_FE*

Attribute Summary	
	N/A

Operation Summary	
Default constructor	Public BPD_FE() N/A
Void	Public myMain() Data flow control

Operation Details	
Default constructor	Public BPD_FE() N/A
Void	Public myMain() The method creates Document objects. Each Document object has one file name as the only initialization parameter. This method also checks the database or GJXDM to make sure no duplicate document ID (control number). One CollectionBuilder object is created. All Document objects are given to the CollectionBuilder object for preprocessing. The results become the segmentWithPOSList in the Document objects. One FeatureDiscover object is created. According to the system configuration, the method dynamically binds either a SerialFeatureExtractor object or a ParallelFeatureExtractor object to FeatureExtractor object. One Document object is given to the extractFeatures() method of the FeatureExtractor object each time to apply information extraction RRE rules to extract features from the Document object. The results are in the allFeatures attribute of the Document object. One ODBC is created. All Document objects are passed to it so that the features extracted and the original text could be stored in the BPD_IE database. One GJXDMConnector object is created. All Document objects are passed to it so that the features extracted and the original text could also be stored in the GJXDM file. The BPD_FE:myMain flowchart contains details.

11.2 *BPD_IE*

Attribute Summary	
Vector<String>	Protected allFileNamesWithPath Each String is a file's name with correct path. These file names are from the retrieveAllFileNames() method

String	Protected configurationFileName This is the file name of the system configuration file. The default value is "BPD_IE.cfg"
Configuration	Protected systemConfig this attribute contains all configuration settings of the system

Operation Summary	
constructor	Public BPD_IE (int argc, char* argv) This is the start point of the system.
bool	Private retrieveAllFileNames(char* argv) Convert input parameters into file names with path.
bool	Private readConfigurations() Read all configurations from the "configurationFileName" file into the systemConfig attribute

Operation Details	
Void	Public BPD_IE (int argc, char* argv) Call readConfigurations() to read all system configurations into systemConfig File names or folder names are input parameters. This method will call retrieveAllFileNames() to obtain all available files indicated by the parameters.
bool	Private retrieveAllFileNames(vector<String> filenames) The input is a list of filenames or fold names. If a parameter is a folder, then get all file names under the folder. After that, push all file names into allFileNamesWithPath If all file names exist, then return true. Otherwise, return false. This method could only be called within the main() function.
bool	Private readConfigurations() Open the configuration file. Read all configurations into corresponding field of the systemConfig. Close the configuration file. If the configuration file does not exist, return false. Otherwise, return true. If some configuration fields are missing in the configuration file, then the default values of these fields are used.

11.3 *BPD_MB*

Attribute Summary	
	N/A

Operation Summary	
Default constructor	Public BPD_MB() N/A
Void	Public myMain() Data flow control

Operation Details	
Default constructor	Public BPD_MB() N/A
Void	Public myMain() Declare ModelBuilder with training parameters Get all rules learned in the Model Builder Save the rules through Storage

11.4 *BPD_MO*

Attribute Summary	
	N/A

Operation Summary	
Default constructor	Public BPD_MO() N/A
Void	Public myMain() Data flow control
Void	Private displayUserInterface (void) Prepare the search form
void	Private fillOutForm (void) Using Document objects' features to fill out corresponding fields on the search form.
bool	Private matchMO (void) Match the search form/keyword to documents in the database
bool	Private ExtractFeature (void) Call FeatureDiscover to extract features for the fillOutForm() method

Operation Details	
Default constructor	Public BPD_MO() N/A
Void	Public myMain() Call displayUserInterface() Call ExtractFeature() Call fillOutForm() Call matchMO() Display search result in summary Display the entire document if necessary
Void	Private displayUserInterface (void) Display the search form, and provide the most often used featureTypes/values.
void	Private fillOutForm (void) Use Document objects' features to fill out corresponding fields on the search form.
bool	Private matchMO (void) Convert the search form/keyword to search SQL commands.

	Search the BPD_IE database. Sort results. Display the results to the user If the search keyword is illegal, or there are database related errors, then return false. Otherwise, return true.
bool	Private ExtractFeature (void) Call FeatureDiscover to extract features All features are stored in a Document object The Document object will be used in the fillOutForm() method If at least one feature on the search form is extracted, then return true. Otherwise, return false.

11.5 *BPD_TD*

Attribute Summary	
	N/A

Operation Summary	
constructor	Public BPD_TD (void) N/A
Void	Public MyMain(void) Data flow control

Operation Details	
constructor	Public BPD_TD (void) N/A
Void	Public MyMain(void) Load Documents Declare Collection Builder Declare TrainingSet Developer Save results into Storage

11.6 *BPDFeatureExtractor*

This is a component of TMI

11.7 *CollectionBuilder*

Attribute Summary	
Enum dataSourceType	Private dataSource This attribute is initialized in the constructor using the system configuration.
String	Private segmentationRuleLocation This is the location of the segmentation rules according to the value of dataSource. This attribute is initialized in the constructor using the system configuration.
enum POSTaggerType	Private POSTagger This attribute decides which part-of-speech tagger should be used in

	the Collection Builder. This attribute is initialized in the constructor using the system configuration.
Vector<String>	Private segmentationRuleList All segmentation rules are stored in the vector. All segmentation rules are read from the "segmentationRuleLocation".

Operation Summary	
constructor	Public CollectionBuilder (Enum dataSourceType dataSourceFromConfig, String segmentationRuleLocationFromConfig, enum POSTaggerType POSTaggerFromConfig) Initialize the attributes of the CollectionBuilder
bool	Private readSegmentationRules() Read in all segmentation rules.
bool	Public splitSegment (Document &currentDocument) Split the plain text of the current document into segments using segmentationRuleList
bool	Public tagPOS (Document &currentDocument) Assign part-of-speech tag to each word in the segments of the currentDocument

Operation Details	
constructor	CollectionBuilder (Enum dataSourceType dataSourceFromConfig, String segmentationRuleLocationFromConfig, enum POSTaggerType POSTaggerFromConfig) The three parameters of the functions are from the system configuration. dataSource = dataSourceFromConfig segmentRuleLocation = segmentRuleLocationFromConfig POSTagger = POSTaggerFromConfig Call readSegmentationRules()
bool	Private readSegmentationRules() Open the segmentation rule connection according to dataSource and segmentRulelocation. Read all rules into segmentationRuleList. Close the connection. If the segmentation connection does not exist, or it is empty, then return false. Otherwise, return true. This method can only be called in the CollectionBuilder constructor.
bool	Public splitSegment (Document &currentDocument) Call the TMI with originalText for sentence boundary detection to split the sentences returned from the TMI into segments, which are stored in the segmentList.segmentText attribute of the currentDocument. The segmentation rules are applied one by one to find the segment boundaries. Each string between any two boundaries becomes a segment

	If the originalText of the currentDocument is empty, then return false. Otherwise, return true. The currentDocument is also returned because it is a reference parameter.
bool	Public tagPOS (Document &currentDocument) Call the CCTagger in TMI for part-of-speech tagging according to the value of POSTaggerType. All segments with part-of-speech tags are stored in the attribute segmentList.wordWithPOSList of the currentDocument. If the size of segmentList is empty, then return false. Otherwise, return true. The currentDocument is also returned because it is a reference parameter.

11.8 *Configuration*

Attribute Summary	
Enum dataSourceType	Public dataSource The source of the segmentation rules: BPD_IE database, a GJXDM file, or a local text file.
String	Public segmentationRuleLocation This setting works together with dataSource. dataSource = GJXDM: this contains the section name of the segmentation rule set that is used in the Collection Builder. It has default value "defaultSegmentRules". dataSource = ODBC: this contains the table name of the segmentation rule set. It has default value "defaultSegmentRules". dataSource = LOCALFILE: this contains the local file name of the segmentation rule set. It has default value "defaultSegmentRules.txt".
Enum POSTaggerType	Public POSTaggingMethod This setting decides which part-of-speech tagger is used in the Collection Builder. It has default value "BrillTagger". This is an enumerated type.
String	Public ODBCname This is the ODBC name for database connection. The default value is a "BPD".
String	Public ODBCuser This is the user name of the ODBC connection. This name must be one of the database user's names. Administrator can set the database user using database administration tool. The default value is "bpdlu".
String	Public ODBCpasswd This is the password of the database user. . The default value is "bpdlu.database".
String	Public RRERulesSource dataSource = GJXDM: this contains the section name of the information extraction RRE rules that are used in the Feature Extractor. The default value is "defaultRRErules". dataSource = ODBC: this contains the table name of the information extraction RRE rules that are used in the Feature Extractor. The

	default value is "defaultRRErules". dataSource = LOCALFILE: this contains the local file name of the information extraction RRE rules that are used in the Feature Extractor. The default value is "defaultRRErules.txt".
Enum FE_Method	Public featureExtractionMethod This configuration determines which feature extractor is used. Whether it is serialFeatureExtractor or parallelFeatureExtractor Enumerated type.
String	Public GJXDM_Filename This attribute contains the GJXDM file name where the system outputs the information extraction results. The default value is "BPD_GJXDM.xml"
String	Public GJXDM_mappingFilename dataSource = GJXDM: This attribute contains section name of the "attribute to GJXDM structure" mappings. The default value is "BPD_GJXDMmappings" dataSource = ODBC: This attribute contains table name of the "attribute to GJXDM structure" mappings. The default value is "BPD_GJXDMmappings" dataSource = LOCALFILE: This attribute contains local file name of the "attribute to GJXDM structure" mappings. The default value is "BPD_GJXDMmappings.txt"
String	Public labelsLocation dataSource = GJXDM: This indicates a section name of the label list in GJXDM. Currently, we only support two labels ("TRUE" and "FALSE"). The default value is "BPD_labels". dataSource = ODBC: This indicates a table name of the label list in GJXDM. The default value is "BPD_labels". dataSource = LOCALFILE: This indicates a local file name of the label list in GJXDM. The default value is "BPD_labels.txt".
String	Public featureTypesLocation dataSource = GJXDM: This indicates a section name of the featureTypes in GJXDM. The default value is "BPD_featureTypes". dataSource = ODBC: This indicates a table name of the featureTypes in GJXDM. dataSource = LOCALFILE: This indicates a local file name of the featureTypes in GJXDM.
String	Public parametersLocation This setting works together with dataSource. dataSource = GJXDM: this contains the section name of the parameters that is used in the Model Builder. It has default value "defaultParameters". dataSource = ODBC: this contains the table name of the parameters used in the Model Builder. It has default value "defaultParameters". dataSource = LOCALFILE: this contains the local file name of parameters used in the Model Builder. It has default value

	"defaultParameters.txt".

11.9 *Document*

Attribute Summary	
String	Protected fileName This is the document's file name with the full path
String	Protected originalText This is the plain text content of the document. The content is directly from the file whose name is the value of fileName.
Vector	Protected segmentList This vector keeps all segments of the document after the segmentation in the Collection Builder. A segment is stored in a Segment structure.
Vector< Feature>	Protected allFeatures This vector contains all features extracted from the current document. A feature is stored in a Feature structure.
String	Protected ControlNumber This is the unique ID of each document. The control number is extracted from the content (originalText) of the document.

Operation Summary	
constructor	Document (String inputFileName) Initialize the fileName and originalText attributes
bool	Private readFile () Read file content into the originalText attribute
String	Public getOriginalText() Return the plain text file's content
Vector	Public getSegmentList() Return all segments
Vector<Feature>	Public getAllFeatures() Return all features extracted from the document
bool	Public setSegmentList (vector segmentListToSet) Assign values to segmentList
bool	Public setAllFeatures (vector< Feature> allFeaturesToSet) Assign values to allFeatures

Operation Details	
constructor	Document (String inputFileName) This constructor takes the inputFileName to initialize the fileName attribute. Next, it calls the readFile() method to read in the plain text content of the file. The content is stored in the originalText attribute.
bool	Private readFile () Open the file using the fileName attribute. Read the file contents into the originalText attribute. Close the file.

	If the file does not exist, or there is read error, then return false. Otherwise, return true. This method can only be called from the Document constructor.
String	Public getOriginalText() Return originalText
Vector<String>	Public getSegmentList() Return segmentList
Vector<Feature>	Public getAllFeatures() Return allFeatures
bool	Public setSegmentList (vector<String> segmentListToSet) segmentList = segmentListToSet If the size of segmentListToSet > 0, then return true. Otherwise, return false.
bool	Public setAllFeatures (vector< Feature> allFeaturesToSet) allFeatures = allFeaturesToSet If the size of allFeaturesToSet > 0, then return true. Otherwise, return false.

11.10 *Element*

Attribute Summary	
String	Public content This is a word or part-of-speech tag
Int	Public gapMin This is the minimum number of characters allowed between the current element and the next element. Note that gapMin=0 when the current element is the last element in the rule/prefix/suffix.
Int	Public gapMax This the maximum number of characters allowed between the current element and the next element. Note that gapMax=0 when the current element is the last element in the rule/prefix/suffix

Operation Summary	
constructor	Public Element (String element, int min, int max) Initialize the element

Operation Details	
constructor	Public Element (String element, int min, int max) Content = element GapMin = min GapMax = max

11.11 *Feature*

Attribute Summary	
String	Public featureType A featureType is the attribute of the feature to be extracted. For

	example, a Height, Crime Date, Location, Weapon, etc.
String	Public featureContent Returns the actual feature. For example, "5 feet six inches" is the featureContent for the featureType Height.
Long	Public Offset_Start The distance from the beginning of the originalText to the start point of the current feature. The number is counted using characters, including the new line character.
Long	Public Offset_End The distance from the beginning of the originalText to the last character of the current feature. The number is counted using characters, including the new line character.
String	Public extractionRuleFired This is the RRE rule used to extract the current feature.

11.12 *FeatureDiscover*

Attribute Summary	
String	Private RRERulesSource This is the location of the information extraction RRE rules. This attribute is initialized in the constructor using the system configuration.
Vector<String>	Protected extractionRRERuleList This contains all RRE rules read from the RRERulesFileName.

Operation Summary	
constructor	FeatureDiscover (String RRERulesSourceFromConfig) Initialize the FeatureExtractor
bool	Private readExtractionRules () Read in all RRE rules from RRERulesSource and store all RRE rules in extractionRRERuleList.
void	virtual Public extractFeatures (vector<Document> &documentList) This a virtual method used for dynamic binding.

Operation Details	
constructor	FeatureExtractor (String RRERulesSourceFromConfig) RRERulesSource = RRERulesSourceFromConfig Call readExtractionRules()
bool	Private readExtractionRules () Open the source data storage using RRERulesSource. Read all RRE rules into extractionRRERuleList. If the RRERulesSource does not exist or there is no rule in it, then return false. Otherwise, return true.
Vector<Document>	virtual Public extractFeatures (vector<Document> &documentList)

11.13 *GJXDMconnector*

Attribute Summary This is a sub class of Storage	
String	GJXDM_Filename This is the name of the GJXDM file that stores all documents with features extracted.
String	GJXDM_mappingFilename This is the name of the GJXDM mapping file that contains all "attribute to GJXDM structure" mappings.
Map<String, String>	GJXDMmapping The key of the map is, but not limited to, featureTypes such as Height. The value of the map is the value of the featureContent such as "6 feet".

Operation Summary	
Construction	GJXDMconnector (String GJXDM_FilenameFromConfig, String GJXDM_mappingFilenameFromConfig) Initialize the GJXDMconnector object.
bool	Private readGJXDMmapping () Read the "attribute to GJXDM structure" mapping from GJXDM_mappingFilename.
bool	Public insertDocument (vector<Document> documentList) Insert all document objects into the GJXDM file
bool	Public insertDataSet (Map<pair<FeatureType, Label>, vector> dataSet) Insert the training dataset developed in BPD_TD into the GJXDM file.
Map<pair<FeatureType, Label>, vector> dataSet	Public readDataSet (void) Retrieve the training dataset from the GJXDM file.

Operation Details	
Construction	GJXDMconnector (String GJXDM_FilenameFromConfig, String GJXDM_mappingFilenameFromConfig) GJXDM_Filename=GJXDM_FilenameFromConfig GJXDM_mappingFilename=GJXDM_mappingFilenameFromConfig Call readGJXDMmapping()
bool	Private readGJXDMmapping () Open GJXDM_mappingFilename Read each line as a mapping Convert the line into the map structure Close GJXDM_mappingFilename If the file does not exist or the file is empty or the format of any line is incorrect, then return false. Otherwise, return true.

bool	Public insertDocument (vector<Document> documentList) Open GJXDM_Filename Call readGJXDMmapping() Convert the document objects into correct format If the GJXDM_Filename does not exist, then create it. If the GJXDMmapping does not have the current featureType as a key, then write "Attribute mapping does not exist" to errorMessage. Save the document into the GJXDM_Filename file. Close GJXDM_Filename If the errorMessage is empty, then return true. Otherwise, return false.
bool	Public insertDataSet (Map<pair<FeatureType, Label> , vector> dataSet)
Map<pair<FeatureType, Label>, vector> dataSet	Public readDataSet (void)

11.14 *Label*

Attribute Summary	
String	Public labelName Currently we have only "True" and "False" values for labelName
String	Public labelDescription The meaning of the label
Int	Public labelID This is the unique ID for each label. The ID is used to compare whether labels are the same or not.

11.15 *ModelBuilder*

Attribute Summary	
Vector	Private trueSet This is the trueSet for the rule discovery algorithm This set is from the dataSet attribute of RREGenerator
Vector	Private falseSet This is the falseSet for the rule discovery algorithm This set is from the dataSet attribute of RREGenerator
Vector<Rule>	Private ruleList All rules learned are saved in this attribute.
String	Private parametersLocation Where the parameters are stored.
Parameter	Private trainingParameters This is all parameters used in RRE discovery.

Operation Summary

constructor	Public ModelBuilder Initialize the ModelBuilder
bool	Private readTrainingParameters (void) Load all training parameters into the "trainingParameters" attribute.
double	Private scoringFunction (int TP, int FP, int FN, int TN) Calculate the current score.
double	Private findRoot (String & root) Find the first element in the RRE, and return its score.
double	Private andLearningProcess (Rule & RRE) Extend the RRE
double	Private gapLearningProcess (Rule & RRE) Narrow the gaps between elements
Vector<Rule>	Public getRuleList (void) Return the RRE rules learned

Operation Details	
constructor	Public ModelBuilder (String parametersLocationFromConfig, map<pair<FeatureType,Label>, vector<String> dataSet) parametersLocation = parametersLocationFromConfig trueSet = the vector<String> of <FeatureType, "TRUE"> falseSet = the vector<String> of <FeatureType, "FALSE">
bool	Private readTrainingParameters (void) Read all parameters into the "trainingParameters" attribute. according to parametersLocation
double	Private scoringFunction (int TP, int FP, int FN, int TN) Calculate the current score. This could be accuracy or F-measure, etc.
double	Private findRoot (String & root) Find the first element in the RRE, and return its score.
double	Private andLearningProcess (Rule & RRE) Extend the RRE using the "AND" Learning Process of the semi-supervised RRE discovery algorithm.
double	Private gapLearningProcess (Rule & RRE) Narrow the gaps between elements using the Gap Learning Process of the semi-supervised RRE discovery algorithm.
Vector<Rule>	Public getRuleList (void) Return ruleList

11.16 *ODBC*

Attribute Summary This is a sub class of Storage	
String	Public ODBCname This is the ODBC name from the configuration
String	Private ODBCuser This is the ODBC user name from the configuration
String	Private ODBCpasswd

	This is the ODBC user's password from the configuration

Operation Summary	
Construction	ODBC (String ODBCnameFromConfig, String ODBCuserFromConfig, String ODBCpasswdFromConfig) Initialize the ODBC object
bool	Public insertDocument (vector<Document> documentList) Insert all document objects into the BPD_IE database
bool	Public insertDataSet (Map<pair<FeatureType, Label>, vector> dataSet) Insert the training dataset developed in BPD_TD into the BPD_IE database.
Map<pair<FeatureType, Label>, vector> dataSet	Public readDataSet (void) Retrieve the training dataset from the BPD_IE database.
Void	Public searchMO (String keywords) Search MO in the BPD_IE database according to keywords.

Operation Details	
Construction	ODBC (String ODBCnameFromConfig, String ODBCuserFromConfig, String ODBCpasswdFromConfig) ODBCname = ODBCnameFromConfig ODBCuser = ODBCuserFromConfig ODBCpasswd = ODBCpasswdFromConfig
bool	Public insertDocument (vector<Document> documentList) Convert the document objects into correct format Prepare the SQL for database operations Connect ODBC Insert the document into the BPD_IE database Write the error message into errorMessage if any If the errorMessage is empty, then return true. Otherwise, return false.
bool	Public insertDataSet (Map<pair<FeatureType, Label>, vector> dataSet)
Map<pair<FeatureType, Label>, vector> dataSet	Public readDataSet (void)
Void	Public searchMO (String keywords)

11.17 *Pair*

Template <class FIRSTCLASS, class SECONDCLASS>

Attribute Summary

FIRSTCLASS	Private _first
SECONDCLASS	Private _second

Operation Summary	
constructor	Public Pair (FIRSTCLASS first, SECONDCLASS second) Initialize _first and _second
FIRSTCLASS	Public getFirst(void)
SECONDCLASS	Public getSecond (void)
bool	Public operator == (pair toCompare)
Operation Details	
constructor	Public Pair (FIRSTCLASS first, SECONDCLASS second) _first = first _second = second
FIRSTCLASS	Public getFirst(void) Return (_first)
SECONDCLASS	Public getSecond (void) Return (_second)

bool	Public operator == (pair toCompare)
	if ((toCompare.getFirst() == _first) && (toCompare.getSecond()
	== _second)) {
	return true;
	}else {
	return false;
	}

11.18 *ParallelFeatureExtractor*

Operation Details	
Vector<Document>	Public Abstract extractFeatures (vector<Document> &documentList) Declare a TMI BPDFeatureExtractor object For each Document object's segments, fork a thread In each thread, call BPDFeatureExtractor.extractTaggedSet() to extract all features. After all Documents' objects are finished, features are stored in each Document object's allFeatures attribute. Return the documentList with features extracted.

11.19 *Parameter*

Attribute Summary	
Int	Public minCoverage
Double	Public BetaOfRoot
Double	Public BetaOfOther
String	Public gapWordChr
Int	Public maxGap
Int	Public regularExpressionLength

Double	Public minScore

11.20 *Rule*

Attribute Summary	
String	Public featureType The rule only works on this special featureType. This value indicates what features the rule can extract.
String	Public specialUser This is the document author's unique identifier.
Vector<Element>	Public prefix This is the context preceding the feature to be extracted. The context is expressed as a reduced regular expression.
Vector<Element>	Public rule This is the reduced regular expression used to extract the exact features.
Vector<Element>	Public suffix This is the context following the feature to be extracted. The context is expressed as a reduced regular expression.
Int	Public tier This is the level of the current rule. The smaller the tier, the more accurate the rule. Tier 1 rules can be directly applied. Tier 2 rules and above are reserved for future use.

Operation Summary	
constructor	Rule N/A
String	GetRuleString (void) Convert the rule structure into a reduced regular expression

Operation Details	
constructor	Rule N/A
String	GetRuleString (void) Convert individual elements into a single reduced regular expression string.

11.21 *RREGenerator*

Attribute Summary	
map<pair<FeatureType,Label>,vector>	Protected dataset This is the training set used in the TrainingSet Developer
Enum dataSourceType	Private dataSource Data source: e.g., BPD_IE database, GJXDM xml file or a local text file. This comes from the configuration.
String	Private labelsLocation

	This value comes from the configuration. dataSource = GJXDM: This indicates a section name of the label list in GJXDM. dataSource = ODBC: This indicates a table name of the label list in GJXDM. dataSource = LOCALFILE: This indicates a local file name of the label list in GJXDM.
Vector<Label>	Protected labelList This attribute keeps all labels such as TRUE and FALSE.
String	Private featureTypesLocation This value comes from the configuration. dataSource = GJXDM: This indicates a section name of the featureTypes in GJXDM. dataSource = ODBC: This indicates a table name of the featureTypes in GJXDM. dataSource = LOCALFILE: This indicates a local file name of the featureTypes in GJXDM.
Vector<String>	Private featureTypeList This attribute stores all featureTypes.

Operation Summary	
constructor	Public RREGenerator (Enum dataSourceType dataSourceFromConfig, String labelsLocationFromConfig, String featureTypesLocationFromConfig) Initialize private attributes
bool	Private loadLabels (void) Read all labels from the labelsLocation.
bool	Private loadFeatureTypes (void) Read all attributes from the attributeLocation.
Map<pair<FeatureType, Label>, vector>	Public getDataSet (void) return the dataset
Void	Public setDataSet(Map<pair<FeatureType, Label>, vector>, currentDataSet) initialize the dataSet

Operation Details	
constructor	Public RREGenerator (Enum dataSourceType dataSourceFromConfig, String labelsLocationFromConfig, String featureTypesLocationFromConfig) dataSource = dataSourceFromConfig labelsLocation = labelsLocationFromConfig featureTypesLocation = featureTypesLocationFromConfig
bool	Private loadLabels (void) The labels are stored in labelList. If the location does not exist, or there are no labels in it, then return

	false. Otherwise, return true.
bool	Private loadFeatureTypes (void) All attributes are stored in attributeList. If the location does not exist, or there are no feature types in it, then return false. Otherwise, return true.
Map<pair<FeatureType, Label>, vector>	Public getDataSet (void) return dataSet
Void	Public setDataSet(Map<pair<FeatureType, Label>, vector>, currentDataSet) dataSet = currentDataSet

11.22 *Segment*

Attribute Summary	
String	Public segmentText This is the content of the segment
Long	Public Offset_start This is the distance from the beginning of the originalText to the start point of the current segment. The number is counted using characters, including the new line character.
Long	Public Offset_end This is the distance from the beginning of the originalText to the last character of the current segment. The number is counted using characters, including the new line character.
TaggedSet	Public wordWithPOSList This is the result from the TMI::tagger

11.23 *SerialFeatureExtractor*

Operation Details	
Vector<Document>	Public Abstract extractFeatures (vector<Document> &documentList) Declare a TMI BPDFeatureExtractor object For each Document object's segments, call BPDFeatureExtractor.extractTaggedSet() to extract all features. This is a "for" loop. All features are stored in each Document object's allFeatures attribute. Return the documentList with features extracted

11.24 *Storage*

Attribute Summary	
String	Protected errorMessage This contains the return error message from the database, GJXDM, or other data store. If there is no error, then errorMessage is empty.

Operation Summary	
constructor	Public Storage errorMessage = ""
bool	virtual Public insertDocument (vector<Document> documentList) This is a virtual method used for dynamic binding
bool	virtual Public insertDataSet (Map<pair<FeatureType, Label>, vector> dataSet) This is a virtual method used for dynamic binding
Map<pair<FeatureType, Label>, vector> dataSet	virtual Public readDataSet (void) This is a virtual method used for dynamic binding
Void	virtual Public searchMO (String keywords) This is a virtual method used for dynamic binding
String	Public getErrorMessage() Return errorMessage

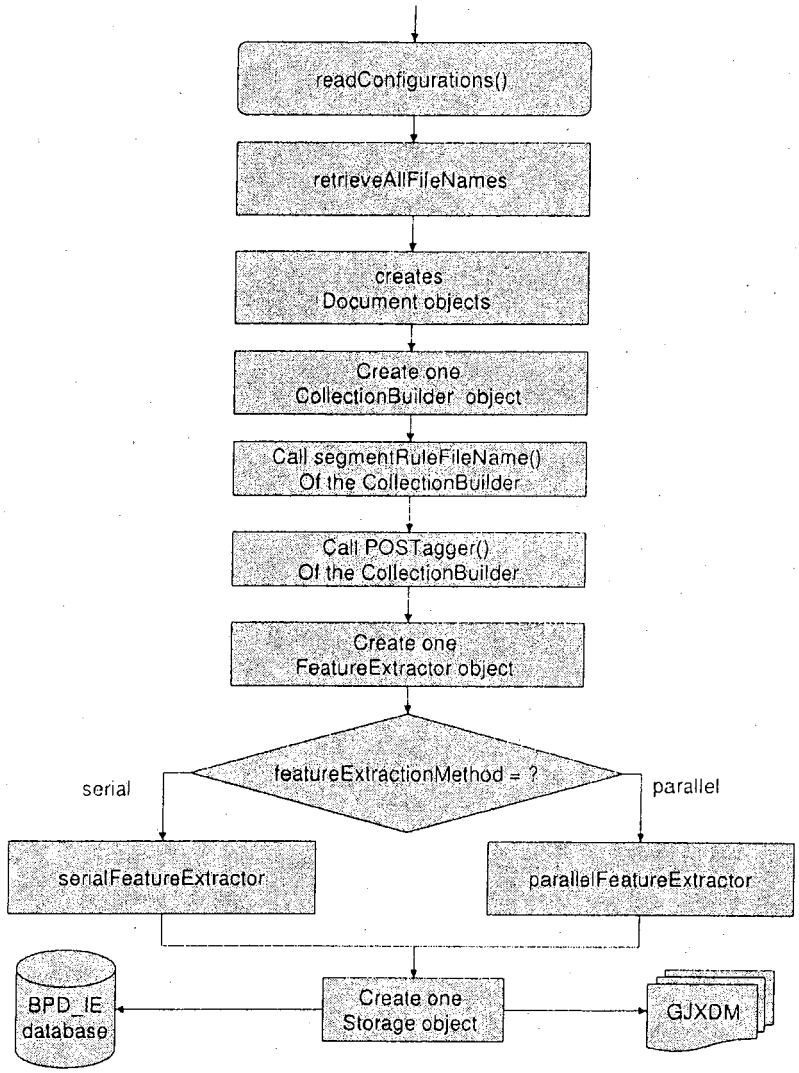
11.25 *TrainingSetDeveloper*

Operation Summary	
Constructor	Public TrainingSetDeveloper (void) N/A
Void	labelSegments (vector <Document>) Display a user interface to let user to label segments

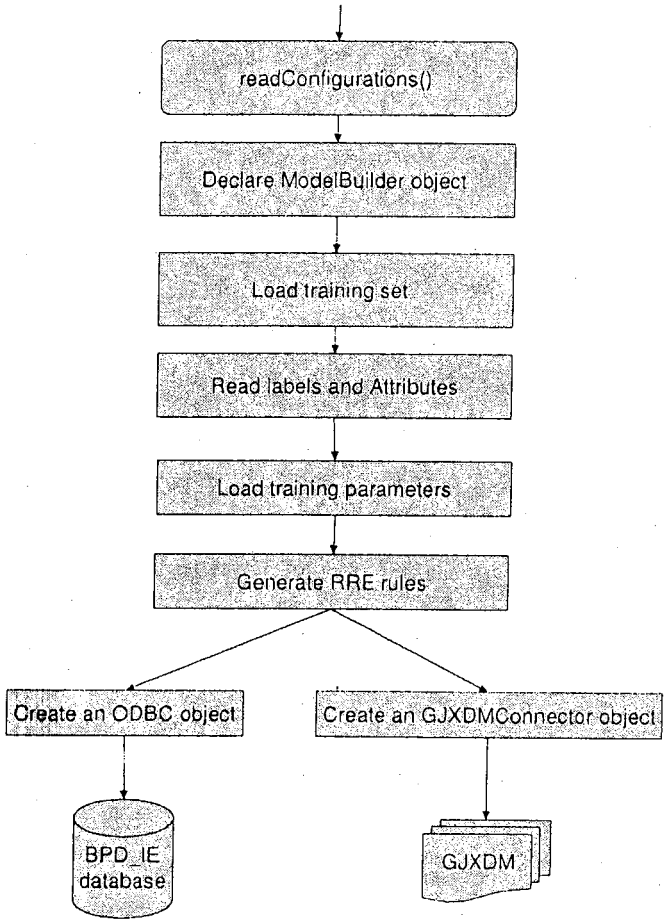
Operation Details	
Constructor	Public TrainingSetDeveloper (void) N/A
Void	labelSegments (vector <Document> & documents) Display a user interface to let user to label segments. All featureTypes, labels, and segments in the documents are displayed. The segments labeled are stored in Document objects.

Appendix E – Flowcharts

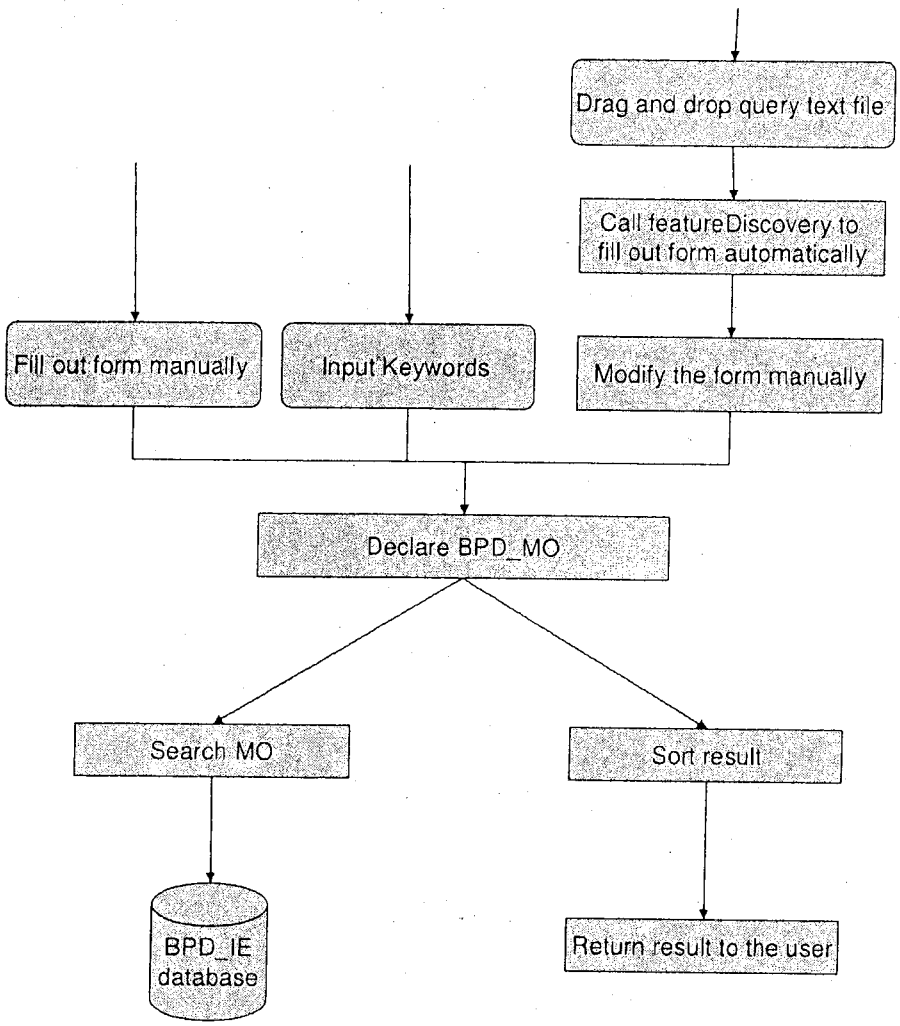
11.26 *BPD_FE:myMain (int argc, char* argv)*



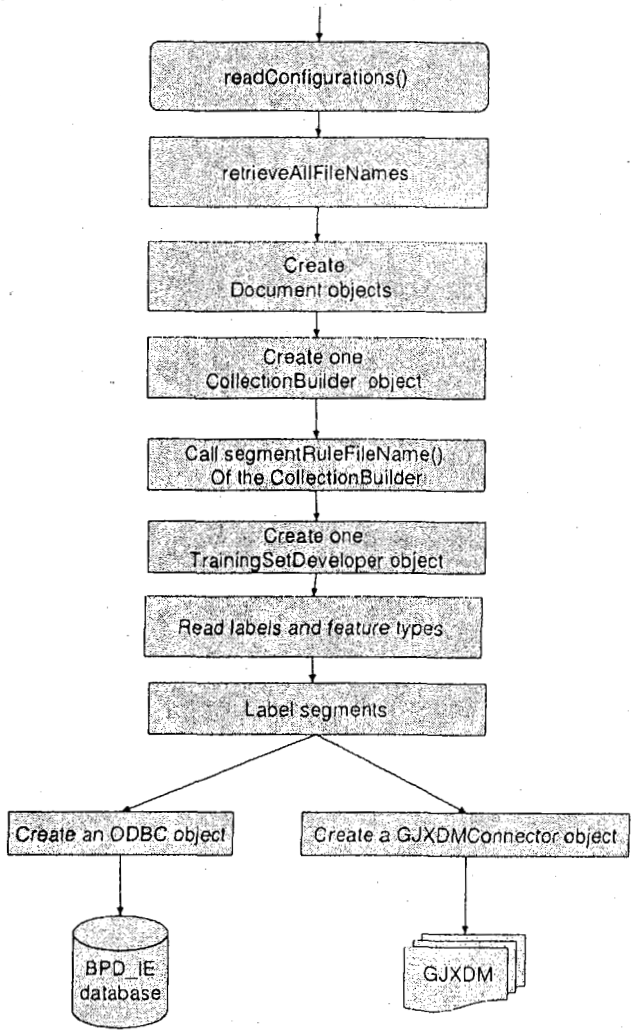
11.27 *BPD_MB:myMain (int argc, char* argv)*



11.28 *BPD_MO:myMain (int argc, char* argv)*



11.29 *BPD_TD:myMain (int argc, char* argv)*



Appendix F – Enumerated Type List

Enum dataSourceType
{ GIXDM, ODBC, LOCALFILE};

Enum POSTaggerType
{ BRILLTAGGER, MAXENTROPYTAGGER };

Enum FE_Method
{ SERIAL, PARALLEL };

Appendix G – Bethlehem Police Department Forms

BETHLEHEM P. D. – ARREST RECORD **FILE NUMBER:**

OFFENSE NUMBER(S) CLEARED	ARREST NUMBER	☐ARREST ☐NON-ARREST	OTN
NAME: LAST FIRST MIDDLE		ALIAS / NICKNAME / MAIDEN NAME	
RESIDENCE: NUMBER & STREET CITY STATE ZIP			PHONE NUMBER(S)

DATE OF BIRTH	AGE	SOCIAL SECURITY NUMBER	SEX	RACE	HEIGHT	WEIGHT	HAIR	EYES

PLACE OF BIRTH: CITY, STATE, COUNTRY	COUNTRY OF CITIZENSHIP	MARITAL STATUS	SPOUSE'S NAME & ADDRESS
DRIVER LICENSE NUMBER & STATE	EMPLOYER / SCHOOL		OCCUPATION / GRADE
EMPLOYER / SCHOOL ADDRESS	PHONE NUMBER (EXTENSION)	MILITARY SERVICE ☐YES ☐NO BRANCH:	
OTHER DESCRIPTIVE INFORMATION (SCARS, MARKS, TATTOOS, AMPUTATIONS, ETC... Describe)			
FATHER'S NAME, ADDRESS & PHONE	MOTHER'S MAIDEN NAME, ADDRESS & PHONE		

DETAILS OF ARREST

DATE AND TIME OF OFFENSE	LOCATION OF OFFENSE	DISTRICT JUSTICE OF OFFENSE
DATE AND TIME OF ARREST	LOCATION OF ARREST	COUNTY
CHARGES:		
ADDITIONAL CHARGES:	PRIOR ARREST ☐YES ☐NO	ARRESTING AGENCY

UNDER INFLUENCE OF: ☐YES ☐ALCOHOL ☐NO ☐NARCOTICS	ARMED ☐YES ☐NO	ILL / INJURED ☐YES ☐NO	MEDICAL AID REQUIRED ☐YES ☐NO	MEDICAL FACILITY	DESCRIPTION OF INJURY / ILLNESS
ACCOMPLICES / CO-CONSPIRATORS					GANG AFFILIATION ☐YES, GANG NAME: ☐NO

ARREST: ☐CITATION ☐WARRANT ☐SUMMONS ☐WRIT WARRANT ☐JUVENILE	CITATION NUMBER(S)	FINGERPRINTED BY	PHOTOGRAPHED BY	NCIC SEARCH ☐YES ☐NO RECORD ☐NO ☐WANTED	
VEHICLE MAKE	MODEL	YEAR	COLOR	REG # & STATE	
VEHICLE OWNER AND ADDRESS			VIN		
VEHICLE DISTINGUISHING CHARACTERISTICS					

ARRAIGNMENT INFORMATION

DATE & TIME	ARRAIGNING DISTRICT JUSTICE	BAIL AMOUNT $	RESULT ☐RELEASED ON BAIL ☐R. O. R. ☐COMMITTED IN DEFAULT ☐COMMITTED WITHOUT BAIL
COMMITTED TO	RELEASED TO	REFERRED TO	

JUVENILE INFORMATION

PARENT / GUARDIAN CONTACTED BY	NAME OF PERSON CONTACTED			RELATIONSHIP TO JUVENILE	
DATE CONTACTED	TIME CONTACTED	CONFERENCE HELD ☐YES ☐NO	CONFERENCE DATE & TIME	HOME STATUS OF PARENTS	
CHARGE ☐ADMITS ☐DENIES	ATTITUDE OF JUVENILE	ATTITUDE / CONDITION OF PARENTS		RELEASED TO / RELATIONSHIP	
SIBLINGS: NAMES & AGES					CHARGED AS ADULT ☐YES ☐NO
FINAL COURT DISPOSITION					RECORDS USE ONLY:
ARRESTING OFFICER(S)	INTERVIEWING OFFICERS:		ASSISTING OFFICERS:		

ADDITIONAL REMARKS / COMMENTS ON BACK

BPD 2 REV. 8/16/2002

BETHLEHEM P. D. – ARREST RECORD FILE NUMBER:

WARNING INDICATORS

- ☐ HAS HAD VIOLENT TENDENCIES
- ☐ HAS BEEN ARMED AND DANGEROUS
- ☐ HAS THREATENED PUBLIC OFFICIALS
- ☐ HAS HAD SUICIDAL TENDENCIES
- ☐ HAS HAD MENTAL DISABILITY
- ☐ HAS RESISTED ARREST
- ☐ INFECTIOUS DISEASE PRECAUTIONS

DOMESTIC VIOLENCE RELATED

☐ YES ☐ NO

PHYSICAL DESCRIPTORS AT TIME OF ARREST

GENERAL

- ☐ BAREFOOT
- ☐ CASUAL / SPORTS
- ☐ DIRTY / SLOPPY
- ☐ GOTHIC
- ☐ GRUNGE
- ☐ NEAT
- ☐ NUDE
- ☐ OTHER
- ☐ PARTIALLY CLOTHED
- ☐ UNIFORMED
- ☐ WELL DRESSED

COMPLEXION

- ☐ ACNE
- ☐ DARK
- ☐ FAIR / LIGHT
- ☐ FRECKLED
- ☐ MEDIUM
- ☐ RUDDY

HANDED

- ☐ RIGHT
- ☐ LEFT
- ☐ AMB.
- ☐ UNKNOWN

HAIR

- ☐ AFRO
- ☐ BALD / BALDING
- ☐ BEADS
- ☐ BRAID / PONY TAIL
- ☐ COLLAR
- ☐ CORN ROW
- ☐ CREW CUT
- ☐ DYED
- ☐ DREADLOCKS
- ☐ MEDIUM / NEAT
- ☐ SHOULDER LENGTH
- ☐ STRAIGHT
- ☐ WAVY / CURLY
- ☐ THIN / RECEDING
- ☐ WIG / TOUPEE

FACIAL HAIR

- ☐ GOATEE
- ☐ FULL BEARD
- ☐ HEAVY MUSTACHE
- ☐ THIN MUSTACHE
- ☐ MEDIUM MUSTACHE
- ☐ HEAVY BROW
- ☐ UNSHAVEN
- ☐ LONG SIDEBURNS
- ☐ NONE

SUSPECT WORE

- ☐ SKI MASK
- ☐ STOCKING MASK
- ☐ CAP / HAT
- ☐ MAKEUP MASK
- ☐ GLOVES
- ☐ SUNGLASSES
- ☐ WIG
- ☐ OTHER MASK
- ☐ GLASSES
- ☐ CLOTHING OPPOSITE SEX

REMAKS / COMMENTS:

B / V.C.

UCR/INCIDENT CODE | BETHLEHEM POLICE DEPARTMENT, BETHLEHEM, PA | CONTROL NUMBER

COMPLAINT/INCIDENT REPORT

OFFENSE / INCIDENT DESCRIPTION | DATE OF REPORT | OCCURRED DATE | TIME | DAY

ADMINISTRATIVE REVIEW | DISPATCH | LOCATION OF OCCURRENCE | DIST. ZONE

V VICTIM COMPLAINANT | RACE | SEX | AGE | D.O.B.

LAST NAME (OR FIRM NAME IF BUSINESS) | FIRST | M.I. | RESIDENCE ADDRESS | CITY / STATE

RESIDENCE PHONE | SIGNATURE AND DATE OF PERSON REPORTING | VICTIM'S PLACE OF EMPLOYMENT | BUS. PHONE

ALCOHOL RELATED? YES ☐ NO ☐ DRUG RELATED? YES ☐ NO ☐ STUDENT ? YES ☐ NO ☐ L.U. ☐ MORAVIAN ☐ OTHER

INJURY? YES ☐ NO ☐

I NATURE & LOCATIONS ON BODY | OFFICER OBSERVED YES ☐ NO ☐ | HOSPITAL | DOCTOR / CORONER

NAME OF RELATIVE / PERSON NOTIFIED | RELATIONSHIP TO INJURED | PHONE | NOTIFIED BY WHOM | CONDITION OF INJURED

W ☐ LAST NAME | FIRST | M.I. | RESIDENCE ADDRESS

PR ☐

RESIDENCE PHONE | R/S | AGE | D.O.B. | PLACE OF EMPLOYMENT | BUS. PHONE

PROPERTY DAMAGE? YES ☐ NO ☐ STOLEN? YES ☐ NO ☐

P TYPE PROPERTY AUTO THEFT ☐ | SERIAL NUMBER | TOTAL VALUE

TYPE PROPERTY BICYCLE THEFT ☐ | SERIAL NUMBER | TOTAL VALUE

TYPE PROPERTY | SERIAL NUMBER | TOTAL VALUE

SIGNIFICANT M.O.? YES ☐ NO ☐

MO # OFFENDERS | DIRECTION & HOW APPROACHED / FLED | IMPLEMENT USED | METHOD USED | OTHER RELATED COMPLAINT NUMBERS

EXACT LOCATION OF VICTIM OR PROPERTY | P.O. ENTRY | P.O. EXIT

TYPE PREMISES | OTHER EVIDENCE OR SUSPECT REMARKS TO VICTIM

SUBJECTS - DESCRIBE EACH SUBJECT USING AGE, RACE, HEIGHT, WEIGHT, ANY IDENTIFYING SCARS, MARKS & CLOTHING

S

SUBJECT #1 (INCLUDE AKA INFO) / SS#	SUBJECT #2 (INCLUDE AKA INFO) / SS#	SUBJECT #3 (INCLUDE AKA INFO) / SS#
SUBJECT #1 MAY BE LOCATED AT / PHONE	SUBJECT #2 MAY BE LOCATED AT / PHONE	SUBJECT #3 MAY BE LOCATED AT / PHONE
SUBJECT #1 DESCRIPTION	SUBJECT #2 DESCRIPTION	SUBJECT #3 DESCRIPTION
ARRESTED ☐ CAN BE IDENTIFIED ☐ PREVIOUSLY SEEN ☐	ARRESTED ☐ CAN BE IDENTIFIED ☐ PREVIOUSLY SEEN ☐	ARRESTED ☐ CAN BE IDENTIFIED ☐ PREVIOUSLY SEEN ☐
ALCOHOL RELATED? YES ☐ NO ☐ DRUG RELATED? YES ☐ NO ☐	ALCOHOL RELATED? YES ☐ NO ☐ DRUG RELATED? YES ☐ NO ☐	ALCOHOL RELATED? YES ☐ NO ☐ DRUG RELATED? YES ☐ NO ☐

WEAPON USED OR EXHIBITED? YES ☐ NO ☐ IMPLIED? YES ☐ NO ☐

DW TYPE | DESCRIPTION | HOW USED

VEHICLE DESCRIPTION? SUSPECT ☐ VICTIM ☐ STOLEN? YES ☐ NO ☐

A YEAR | MAKE | MODEL | COLOR | REG# | STATE

VEHICLE REGISTERED OWNER | ADDRESS | CITY / STATE | V.I.N.

VEHICLE TOWED BY | VEHICLE STORED | INSURANCE CO | POLICY # | DISTINCTIVE FEATURES

WEATHER CONDITIONS | PERSONS NOTIFIED

SUNNY ☐ RAIN ☐ SNOW ☐ ICE ☐ FOG ☐ OTHER

ASSISTING OFFICERS | CASE STATUS: ☐ OPEN ☐ ARREST ☐ NON-ARREST ☐ EXCEPTIONAL ☐ INCIDENT ☐ INCIDENT SUPPLEMENT

REPORTING OFFICER	BADGE #	APPROVING SUPERVISOR	BADGE #	INVESTIGATOR	BADGE #
BUREAU ASSIGNMENT					

Appendix H: Importing the Affidavit Database into the BPD_IE Database

1. Importing from the Affidavit Database

All current Affidavit databases are in DBIII format (.dbf) with indexes. Before we can make use of the data, we must convert the DBIII databases into a common database format such as Microsoft Access (.mdb)

The original data is located in "G:\user\Police PRS", and a copy is in "c:\Documents and Settings\Tianhao Wu\Desktop\Police PRS".

To convert, open one of the databases using Microsoft Access as a DBIII file, then save it as an .mdb file.

Note: we have converted all current .dbf files to .mdb files with exactly the same file names except for the file extension.

2. Converting to the BPD_IE MySQL Database

In this section we briefly review the conversion from the original DBIII schema used in the Affidavit database to the schema used in the BPD_IE MySQL database.

2.1. OUBPOENA.mdb & SUBPOENA.mdb

OUBPOENA.mdb includes 40 records & SUBPOENA.mdb includes 58 records.
The two databases have the exact same format.

affidavit_tablename_controlnumber_name is the pattern to use for document names in the DOCUMENTS table. For example, the first record in SUBPOENA.mdb will be saved as affidavit_ SUBPOENA_12345678_John in the BPD_IE DOCUMENTS table, given that the control number for the first record is 12345678 and the person's name is "John".
We use the following three fields in SUBPOENA to depict how a record in this table is converted into record(s) in the BPD_IE tables.

SUBPOENA

Incident	Name	Subp_narr
12345678	John	This is a text description

SUBPOENA is mapped to the following two tables in BPD_IE: each record in SUBPOENA is mapped to one record in the DOCUMENTS table. Each record in SUBPOENA is also mapped to 47 records in the FEATURES table. Note that "Subp_narr" is only in the DOCUMENTS table. See below for an example.

DOCUMENTS

documentName	originalText	localHostName
affidavit_ SUBPOENA_12345678_John	Put Subp_narr's content here: "This is a text description"	Your local computer's name should be put here

FEATURES

featureType	featureContent	startValueSTR	endValueSTR	certainty	startValueNum	endValueNum	documentName	localHostName
Control number	12345678	12345678	12345678	1	12345678	12345678	affidavit_SUBPOENA_12345678_John	Your local computer's name should be put here
Person Name	John	John	John	1			affidavit_SUBPOENA_12345678_John	Your local computer's name should be put here

Additional field to feature type mappings

DATABASE-TABLE	ORIGINAL_FIELD_NAME	BPD_IE_FeatureType_NAME
SUBPOENA	incident	controlnumber
	name	name
	Name1	Name
		resident
	Name1P	phoneNumber
	Emp1	employment
	Emp1p	phoneNumber
	Name2	Name
		resident
	Name2P	phoneNumber
	Emp2	employment
	Emp2p	phoneNumber
	Name3	Name
		resident
	Name3P	phoneNumber
	Emp3	employment
	Emp3p	phoneNumber
	Name4	Name
		resident
	Name4P	phoneNumber

	Emp4	employment
	Emp4p	phoneNumber
	Name5	Name
		resident
	Name5P	phoneNumber
	Emp5	employment
	Emp5p	phoneNumber
	Name6	Name
		resident
	Name6P	phoneNumber
	Emp6	employment
	Emp6p	phoneNumber
	Name7	Name
		resident
	Name7P	phoneNumber
	Emp7	employment
	Emp7p	phoneNumber
	Name8	Name
		resident
	Name8P	phoneNumber
	Emp8	employment
	Emp8p	phoneNumber
	Name9	Name
		resident
	Name9P	phoneNumber
	Emp9	employment
	Emp9p	phoneNumber
	Name10	Name
		resident
	Name10p	phoneNumber
	Emp10	employment
	Emp10p	phoneNumber
	Name11	Name
		resident
	Name11P	phoneNumber
	Emp11	employment
	Emp11p	phoneNumber
	Dateentr	reportdate
	Subp_narr	Document original text

2.2. ooldata.mdb

ooldata.mdb includes 1 record.

The document names used in the DOCUMENTS table are: affidavit_ooldata_deptname

OOLDATA

deptaddr1	dj1phone
Packer lab 117	6107583737

OOLDATA is mapped to the following two tables in BPD_IE: each record in OOLDATA is mapped to one record in the DOCUMENTS table. Each record in OOLDATA is also mapped to 15 records in the FEATURES table. See below for an example.

DOCUMENTS

documentName	originalText	localHostName
affidavit_ooldata_deptname	empty	Your local computer's name should be put here

FEATURES

featureType	featureContent	startValueSTR	endValueSTR	certainty	startValueNum	endValueNum	documentName	localHostName
Location	Packer lab 117	Packer lab 117	Packer lab 117	1			affidavit_ooldata_deptname	Your local computer's name should be put here
PhoneNumber	6107583737	6107583737	6107583737	1	6107583737	6107583737	affidavit_ooldata_deptname	Your local computer's name should be put here

Additional field to feature type mappings

DATABASE-TABLE	ORIGINAL_FIELD_NAME	BPD_IE_FIELD_NAME
OOLDATA	Deptname	person name
	deptaddr1	location
	deptaddr2	location
	dj1name	person name
	dj1addr1	location
	dj1addr2	location
	dj1phone	phone number
	dj2name	person name

	dj2addr1	location
	dj2addr2	location
	dj2phone	phone number
	dj3name	person name
	dj3addr1	location
	dj3addr2	location
	dj3phone	phone number

2.3. Ochwar.mdb & ochwar1.mdb & schwar.mdb

Ochwar includes 73 records. Ochwar1 is exactly the same as ochwar.

Schwar includes 308 records.

The document names used in the DOCUMENTS table are: affidavit_ochwar_incno, affidavit_ochwar1_incno and affidavit _schwar_incno.

OCHWAR

cause	Desc	What	app_date
Paragraph 1	Paragraph 2	Paragraph 3	01/23/2000

OCHWAR is mapped to the following two tables in BPD_IE: each record in OCHWAR is mapped to one record in the DOCUMENTS table. Each record in OCHWAR is also mapped to four records in the FEATURES table. Note that "cause", "desc", and "what" are only in the DOCUMENTS table. See below for an example.

DOCUMENTS

documentName	originalText	localHostName
affidavit_ochwar_incno	Paragraph 1 Paragraph 2 Paragraph 3	Your local computer's name should be put here

FEATURES

featureType	featureContent	startValueSTR	endValueSTR	certainty	startValueNum	endValueNum	documentName	localHostName
report date	01232000	01232000	01232000	1	01232000	01232000	affidavit_ochwar_incno	Your local computer's name should be put here

Additional field to feature type mappings

DATABASE-TABLE	ORIGINAL_FIELD_NAME	BPD_IE_FIELD_NAME
Ochwar	Cause	original text (first paragraph)
	Incno	controlnumber
	off1	person name
	phone1	phone number
	app_date	report date
	Desc	original text (second paragraph)
	What	original text (Third paragraph)