

# Interstate Criminal History Transmission Specification

XML Version 3.00

*Joint Task Force on Rap Sheet  
Standardization*

February, 2005

# Table of Contents

<b>1.0 INTRODUCTION</b> .....	<b>3</b>
1.1 Background.....	3
1.2 Joint Task Force Accomplishments.....	3
1.3 Organization of Specification.....	4
<b>2.0 PUBLIC SAFETY AND THE CRIMINAL HISTORY RECORD: AN ILLUSTRATIVE IMPLEMENTATION OF THE INTERSTATE CRIMINAL HISTORY SPECIFICATION</b> .....	<b>7</b>
2.1 The Importance of the Criminal History Record and a Uniform Standard.....	7
2.2 The Criminal Records Information Exchange System.....	8
2.3 An Illustrative Implementation.....	9
2.4 Transition.....	10
2.5 Record Request.....	10
2.6 Criminal History Transmission to NLETS.....	13
<b>3.0 ELEMENT DICTIONARY</b> .....	<b>14</b>
3.1 Introduction (<rap:Introduction>).....	15
3.2 Person Identification (<j:Subject>).....	19
3.3 Cycle (<rap:Cycle>).....	54
3.4 Agency Index (<j:Organization>).....	79
<b>4.0 RAP SHEET DEFINITION</b> .....	<b>83</b>
4.1 Detailed Structure.....	83
4.2 XML Schemas.....	101
<b>5.0 EXAMPLE RAP SHEET</b> .....	<b>130</b>
<b>APPENDIX A: XML TRANSFORMATION</b> .....	<b>150</b>
5.1 Global Justice XML Data Model.....	150
5.2 Sources.....	152
5.3 <a href="http://it.ojp.gov/topic.jsp?topic_id=109">http://it.ojp.gov/topic.jsp?topic_id=109</a> .....	152
5.4 <a href="http://it.ojp.gov/documents/whatisXML.doc">http://it.ojp.gov/documents/whatisXML.doc</a> .....	152
5.5 <a href="http://it.ojp.gov/jxdm/faq.html#N10079">http://it.ojp.gov/jxdm/faq.html#N10079</a> .....	152
5.6 <a href="http://it.ojp.gov/jxdm/refnotes.html#N10079">http://it.ojp.gov/jxdm/refnotes.html#N10079</a> .....	152
<b>APPENDIX B: XML APPLICATIONS</b> .....	<b>153</b>
<b>APPENDIX C: EXAMPLE STYLE SHEET</b> .....	<b>155</b>
<b>APPENDIX D: TEXT</b> .....	<b>361</b>
<b>APPENDIX F: REFERENCE DOCUMENTS</b> .....	<b>367</b>
<b>APPENDIX G: JOINT TASK FORCE ON RAP SHEET</b> .....	<b>369</b>
<b>APPENDIX H: NATIONAL TASK FORCE ON INCREASING THE UTILITY OF THE CRIMINAL HISTORY RECORD (1993-1995)</b> .....	<b>371</b>

# ***1.0 Introduction***

## **1.1 Background**

In 1995, the National Task Force on Increasing the Utility of the Criminal History Record recommended expanded data content, a presentation format (page layout) for the expanded content, and the creation of a transmission format for the interstate sharing of criminal history information.<sup>1</sup> The National Task Force included representatives from the Federal Bureau of Investigation (FBI), FBI Criminal Justice Information Services Advisory Policy Board (CJIS APB), National Law Enforcement Telecommunications System (NLETS), the National Center for State Courts, SEARCH, the National Consortium for Justice and Statistics. Its members were a diverse array of justice practitioners drawn from the judiciary, prosecution, court administration, local, state, and federal law enforcement, juvenile justice pre-trial services and state criminal records repositories.<sup>2</sup> In 1996, the Joint Task Force on Rap Sheet Standardization with representation from the FBI CJIS Division, the CJIS APB, NLETS, SEARCH and state and local law enforcement agencies, was formed to carry forward the work of the National Task Force by developing a standardized criminal history transmission format.

## **1.2 Joint Task Force Accomplishments**

The Joint Task Force on Rap Sheet Standardization has accomplished three major objectives:

- An XML based standardized criminal history transmission format
- A presentation format utilizing the XML transmission format
- An illustrative example which combines criminal histories from multiple sources into a single criminal history

Implementation of this specification by all states and the FBI benefits the end user of criminal history records by providing information that is more easily understood, more complete and of higher accuracy than was heretofore possible. It will provide a method by which an authorized user who requests an interstate criminal history record, regardless of the request method:

---

<sup>1</sup> See *Increasing the Utility of the Criminal History Record: Report of the National Task Force*

<sup>2</sup> Members of the National Task Force on Increasing the Utility of the Criminal History Record are listed in Appendix G

- Will always<sup>3</sup> receive the *same set of information*
- Will always receive a *single record*<sup>4</sup> for multi-source interstate criminal histories, in which the criminal justice event cycles are presented in date order
- Upon request will receive the record in *computer-readable format*<sup>5</sup> for use in filling display screens, data entry screens or databases, or for editing or state-specific presentation formats
- Upon request will receive the record at an approved destination *whether or not*<sup>6</sup> it is served by an intrastate law enforcement network.

This version of the Interstate Criminal History Transmission Specification, Version 3.0, is based upon the Global Justice XML Data Model 3.0.2. The informational content is largely unchanged from the previous XML versions 2.X, and the ANSI/NIST versions 1.X.

### **1.3 Organization of Specification**

The specification has been divided into five sections and six appendices. The Executive Summary provides a high level description of the XML national standardized rap sheet project. This summary gives the reader an understanding of both the history of the project, the reasons for its creation, and the anticipated benefits. The sections following this detail specifications for the implementation of the XML national rap sheet. A brief description of each section follows:

#### *1.0 Introduction*

This section presents a brief background of the project since its inception.

---

<sup>3</sup> Now, responses to III requests use FBI-held information for non-III states and state data for III and NFF states. Responses to fingerprint submissions use FBI data for both non-III and III state's data. NFF states provide their data to the FBI for inclusion in responses to fingerprint submissions. This specification calls for use of state data for all responses for data held by III-states. The exception is for non criminal justice purpose codes, where use of state-held data is limited to those states that have signed the interstate National Crime Prevention and Privacy Compact or are III/NFF participants.

<sup>4</sup> Now, the user receives several criminal history records if III/NFF state repositories are involved.

<sup>5</sup> The first computer readable format is the transmission format. Others may be developed in the future.

<sup>6</sup> For example, states and/or NLETS may provide facsimile servers and e-mail servers to get the record to the final destination.

- 2.0 *Public Safety and the Criminal History Record: An Illustrative Implementation of the Interstate Criminal History Specification*  
This section provides a general overview of the project and a detailed description of the process for using XML to support the exchange of rap sheets. It also describes how criminal records are merged and delivered.
- 3.0 *Element Dictionary*  
This section describes all of the data elements and their XML tags for the national rap sheet.
- 4.0 *Rap Sheet Definition*  
This section contains the full rap sheet definition including the structure of the rap sheet and the XML schema definition of the root rap sheet element and all its child elements.
- 5.0 *Example Rap Sheet*  
This section presents an example of an XML rap sheet.
- Appendix A XML Transformation*  
This appendix provides a brief primer of the XML conventions adopted for use in defining the rap sheet.
- Appendix B XML Application*  
This appendix provides a technical description of methods to use the XML rap sheet once it has been received by the state. This may include excerpting data and creating a presentation of a rap sheet for display.
- Appendix C XML to Text Style Sheet*  
This appendix provides a set of rules for merging XML transmission format rap sheets from multiple sources into a single, combined rap sheet.
- Appendix D Example Text Rap Sheet*  
The final appendix presents a list of the source materials that were used to create this document.
- Appendix E Content Merging Rules*  
Algorithms and processes are defined for merging XML rapsheets from multiple states into a single rapsheet document.
- Appendix F Reference Documents*  
The National Task Force developed the concept of a presentation format for an interstate rap sheet. Its work provided the foundation for the Joint Task Force on Rap Sheet Standardization. This Appendix identifies the members of the National Task Force.

*Appendix G Task Force Members*

Identifies past and present Joint Task Force members.

## ***2.0 Public Safety and the Criminal History Record: An Illustrative Implementation of the Interstate Criminal History Specification***

### **2.1 The Importance of the Criminal History Record and a Uniform Standard<sup>7</sup>**

Public policy demands background screening of applicants for positions of trust in and outside of government, and volunteers, especially those who work with our most vulnerable populations – children, the elderly, and the disabled. Legislation enacted to strengthen homeland security in the wake of September 11, 2001 expands the types of positions and activities for which background screening, including a criminal history check, is required. This noncriminal justice purpose suitability evaluation enhances public safety by denying jobs and opportunities to those whose criminal history records suggest a potential to do harm or are otherwise unfit.

The criminal history record is central to the effective functioning of the criminal justice system. Research has shown that as many as two-thirds of all persons arrested for criminal offenses have prior criminal records, often including offenses in multiple jurisdictions or States. At every stage of the criminal justice system the criminal history record supports decision making. It is used by the police in many ways, including as an investigative tool and to determine a suspect's current status as a probationer, parolee, or bailee. The presence or absence of a prior criminal record is arguably the most relevant information to a judge or magistrate making a pretrial decision on whether and under what conditions to release a person on bail. Prosecutors use criminal history records from the moment they become involved in criminal cases until the cases are terminated at the defendants' parole hearings or earlier. Courts customarily receive criminal history information in modified form such as in bail reports, or presentence reports prepared by probation departments, or in presentations by the prosecutor. Among the uses of the criminal history record by correctional agencies are inmate classification and making decisions about eligibility for good time credits, early release, work furlough, or release on parole.

---

<sup>7</sup> The source for much of the discussion in Sections 2.1 and 2.2 is Robert R. Belair, Paul A. Woodard, and Eric C. Johnson, *Use and Management of Criminal History Record Information: A Comprehensive Report, 2001 Update*, Criminal Justice Information Policy Series, NCJ 187670 (Washington, D.C.: U.S. Department of Justice, Bureau of Justice Statistics, December 2001).

At the State level criminal history records are collected, maintained, and disseminated by “State central repositories.” These agencies or bureaus within State government are often housed within the State Police, a cabinet-level agency such as the Department of Public Safety or the Attorney General’s Office. Typically, State law requires the repository to establish comprehensive criminal history records and establish rules and regulations for their dissemination to criminal justice and noncriminal justice users. All 50 States, Puerto Rico, and the District of Columbia have established central repositories for criminal history records.

At the Federal level, the FBI is the criminal history information repository for both Federal and foreign offender information and for records of arrests and dispositions forwarded to the FBI from the State records repositories or, to a much lesser extent, from local law enforcement agencies.<sup>8</sup>

A uniform criminal history record format has never been made mandatory. Likewise, no mandatory guidelines regarding the content of criminal history records have ever been promulgated. State and Federal repositories have been left to adopt their own record formats and approaches concerning the types of offenses that should be included on criminal history records and the types of information that should be included. Not surprisingly, this has resulted in considerable diversity in the content and formats of the criminal history records presently generated by the State repositories and the FBI, often leading to difficulty in interpreting the information provided. This confusion is frequently heightened when the information user is from a state other than that which provided the information. Similarly, noncriminal justice users often lack the knowledge and experience to competently interpret the differences in details and layout among the many pieces of information that may surface from a criminal history records check. National adoption of this voluntary Interstate Criminal History Transmission Specification and its associated presentation format, resolves many of the difficulties that hamper the exchange and interpretation of criminal history records.

## **2.2 The Criminal Records Information Exchange System**

The Interstate Identification Index (III) System is an interstate/Federal-State computer network that provides the means of conducting national criminal history record searches to determine whether a person has a record anywhere in the country. It is designed to tie the automated criminal history record databases of State central repositories and the FBI together into a national system by means of an “index-pointer” approach. Here, the FBI maintains an automated master name index, which includes name and identifying data concerning all persons whose automated criminal history records are available by means of the III System.

---

<sup>8</sup> The decision to channel all arrest and disposition information from within a state through the State repository to the FBI is a State decision. States began instituting this practice as early as 1973. As of June, 2002, 49 States, the District of Columbia and Guam are sole-source program participants.



If a search of the III Index indicates that the search subject has an indexed record, the index will “point” the inquiring agency to the FBI and/or to one or more of the State repositories from which the record or records may be obtained. The inquiring agency may then obtain the records directly from the indicated sources by means of the NCIC and the National Law Enforcement Telecommunications System (NLETS). The FBI also maintains the National Fingerprint File (NFF), a database of fingerprints, relating to an arrested or charged individual maintained by the FBI to provide positive identification of record subject indexed in the III System.

The III System is fully operational nationwide. The NCIC system and interfaced State telecommunication systems, together with the FBI’s Integrated Automated Fingerprint Identification System (IAFIS) and computerized State criminal history systems provide III system access to Federal, State, and local criminal justice agencies throughout the country. The system is used to conduct national searches, both name and fingerprint-based, and to provide records responses from FBI and State repository sources for both criminal justice and noncriminal justice purposes. Procedures for conducting searches for criminal justice purposes differ from those for conducting searches for noncriminal justice purposes. **Similarly, the process for responding to inquiries differs** based on whether the request is initiated by a fingerprint submission or whether it is name based, and whether it is for a criminal justice or noncriminal justice purpose

### **2.3 An Illustrative Implementation**

A criminal history will be requested by a criminal justice agency, court or other authorized entity whose state or federal service provider will forward the request to the FBI. Computer terminal requests from authorized users are in the form of an FBI/Interstate Identification Index (III) “QR” transaction. A “QR” is a criminal history record request message. It encompasses several purpose codes that are used to indicate the reason for the inquiry (e.g., criminal justice purpose, security clearance, firearms-related screening, administrative, etc.). After determining the FBI record identification number for the requested information, the FBI will notify each of the states that have III responsibility for a portion (or all) of the criminal history record and NLETS. Each state and the FBI will send their criminal history record to NLETS in transmission format.

When NLETS receives a criminal history in transmission format, it will determine if the record is single or multi-source based on the notification received from the FBI. If it is multi-state, NLETS will hold the portion until all portions are received or until a specified time period has elapsed. NLETS will then assemble the portions into a single, national criminal history. That criminal history will contain all of the information each respondent agency has provided and assembled in such a fashion that the same information received from more than one source is displayed only once. Events are ordered from earliest to latest regardless of the state (or federal authority responsible for the event) in which they occurred.

For both single-state and multi-state criminal histories, NLETS will create a summary record as defined in this specification. Certain fields have been coded so they can be counted. For example, the Charge Severity field is coded so that separate counts can be kept for misdemeanors and felonies and court dispositions are coded to count convictions and non-convictions.

NLETS will then determine the format of the criminal history response. This information may be contained in the original request for the criminal history or may be contained in a preference table maintained by NLETS.

Finally, NLETS will send the response. In most cases, this will be to the state that will forward the response to a user on its state network. Other possibilities may include use of the Internet or fax servers.

## **2.4 Transition**

All states and the FBI will be participants when this Interstate Criminal History Transmission Specification is fully implemented. This may take several years. NLETS will provide the bridge between participating and non-participating states/FBI during the transition period. Throughout the transition period, some states or the FBI may not be ready to send the transmission format and will send a plain text criminal history instead. When NLETS receives such a criminal history, it will simply pass it through without any processing. NLETS will also provide translation between the transmission format and the presentation format contained in the specification, both during and after the transition period.

## **2.5 Record Request**

Requests for rap sheets begin with a message requesting criminal history information to the FBI/III system, or a message to a state criminal history repository or with a fingerprint submission to the FBI. In any case, a national rap sheet may be generated and will be processed by NLETS.

A request for a criminal history record is formulated as a QR transaction containing either the FBI number or a SID (State Identification Number, usually assigned by a state fingerprint identification bureau). Once the request is received by the FBI/III, the state repositories which hold the record portions will be notified electronically via §.A.CHR messages. Each §.A.CHR message describes the record to be retrieved and the destination of the rap sheet. The FBI/III will simultaneously notify NLETS with the same information using the standard notification sent to states, commonly known as the “EL01” and “FL01” messages. These messages contain an index of the state and/or federal records to be merged by NLETS.

FBI/III responds to the requester via the National Crime Information Center (NCIC) telecommunications network with a message indicating that the response is being processed and the type of response the requester should expect:

- A message header beginning with “F” indicates that FBI/III will be the sole respondent or one of multiple respondents, and provides the FBI number and the SID(s) of any state segments. FBI will provide a copy of this message to NLETS.
- A message header beginning with “E” indicates that one or more states will respond but FBI/III will not, and provides the FBI number and the SID(s) of the state segments. FBI will provide a copy of this message to NLETS.
- A message header beginning with “N” indicates that no record matching the QR query terms can be found and processing ceases. There is no need for NLETS to receive this message from the FBI.

An NLETS request for an out-of-state criminal history is triggered by an NLETS IQ or FQ record request transaction, transmitted to the record-holding state via NLETS. Inquiries of this type are inherently single-state queries that do not require response-merging as do multi-state responses. No special handling at the time of inquiry is needed to process such a query.

The FBI type “F” response format is shown in the following example. The critical elements of the message, on which successful processing depends, are shown underscored.

```
FL01HEADERXXXX01234
AKAST0100
THIS INTERSTATE IDENTIFICATION INDEX RESPONSE IS THE RESULT OF YOUR
RECORD REQUEST FOR FBI/901100. INDIVIDUAL'S RECORD WILL BE COMPLETE
WHEN ALL RESPONSES ARE RECEIVED FROM THE FOLLOWING SOURCES EXCEPT FOR
THOSE INDICATED AS DECEASED:
FBI - FBI/901100
NORTH CAROLINA - STATE ID/NC0142585A - DECEASED
TEXAS - STATE ID/TX01346790

END
```

The FBI type “E” response format is shown in the following example. The critical elements of the message, on which successful processing depends, are shown underscored.

```
EL01HEADERXXXX01234
AKAST0100
THIS INTERSTATE IDENTIFICATION INDEX RESPONSE IS THE RESULT OF YOUR
RECORD REQUEST FOR FBI/901100. INDIVIDUAL'S RECORD WILL BE COMPLETE
WHEN ALL RESPONSES ARE RECEIVED FROM THE FOLLOWING SOURCES EXCEPT FOR
THOSE INDICATED AS DECEASED:
NORTH CAROLINA - STATE ID/NC0142585A - DECEASED
```

END

For both type “E” and type “F” responses, the critical format items are as follows. The field AKAST0100, which must appear as the first line, is the requester ORI. If an FBI number preceded by FBI/ is in the narrative block, it will be used to determine whether state records received are part of this merge operation. If an SID number preceded by SID/ is in the narrative block, it — along with the other SID numbers in the record — will be used to determine whether state records received are part of the merge operation. The individual state or FBI responses are compared to these numbers to select records for the merge operations. The processing of responder lines depends on the leading space character, the presence of a space-surrounded hyphen, the STATE ID/ or FBI/ tag, and the optional word DECEASED preceded by a space-surrounded hyphen.

The requester and FBI or SID numbers comprise the unique key for the merge-tracking entry. For this reason, a requester cannot have more than one request for the same FBI or SID numbers active at any one time. If this restriction is violated, results are somewhat unpredictable and will probably result in a single response being issued by NLETS and duplicate responder state rap sheets being ignored.

## 2.6 Criminal History Transmission to NLETS

Regardless of the request method, a time comes when a criminal history is generated and prepared for transmission. In cases where the FBI/III holds the entire record, it will transmit the record using the NLETS network. In cases where one or more states hold the record or portions thereof (FBI/III may also hold a part of the record), each holder will transmit its portion to the requester using the NLETS network. If the requesting state also holds a record on the subject, it may opt to send its record to NLETS to be merged with the other records.

To facilitate this, the FBI will include, in all EL01 and FL01 messages, a listing of all states that hold a record on the subject rather than just a notification that the requesting state also may have a record on the subject. In these instances, the state of inquiry will also receive a \$.A.CHR message. This is necessary so that the state can transmit its response to NLETS so that the record-merge operation described in Appendix E can take place.

Rap sheets sent by the holder in any format other than the transmission format are sent as normal text-based NLETS messages and go directly to the requester without any merge processing at NLETS.

Criminal histories sent in the Interstate Criminal History Transmission Specification format in response to a III QR transaction and IQ or FQ requests carry the transaction type of STDRAP.

The notification messages from the FBI/III to NLETS described in this section cause database entries to be made that control merge processing described in Appendix C. Because of transmission delays, rap sheets may arrive at NLETS prior to the notification message being processed. In this case, the early rap sheets are sent directly to the requesting agency without merging.

## 3.0 Element Dictionary

The element dictionary defines the contents of the elements in the standard Rap Sheet document. In general, if an element is *optional* and the sender has no data for that element, then the element should not be transmitted. When the sender has no data for a *required* element, the word "Unknown" is the preferred value, as in these examples:

```
<j:Organization>
  <j:OrganizationName>Unknown</j:OrganizationName>
</j:Organization>

<j:PersonDriverLicenseID>
  <j:IDIssuingAuthorityText>Unknown</j:IDIssuingAuthorit
yText>
  <j:ID>C4556289248R</j:ID>
</j:PersonDriverLicenseID>
```

Where data elements conform to the Global Justice XML Data Model (GJXDM), definitions have been copied from the model and appear in this format:

**GJXDM:** A number issued by the FBI's Automated Fingerprint Identification System (AFIS) based on submitted fingerprints.

These definitions have been reproduced from GJXDM version 3.0.1. See the current version for most recent definitions at <http://justicexml.gtri.gatech.edu>. Namespace references for elements conforming to GJXDM in this specification appear in this format: <j:>. Namespace references for elements that do not conform to GJXDM appear in this format: <rap:>.

The <rap:RapSheet> element itself has three required attributes: @reportedDate, @xmlns, and @version. The @reportedDate is the date the rapsheet was created. The format for the date is CCYY[-MM[-DD]]. A standard XML name space must be specified in the xmlns attribute using the value printed in the example below. This gives a reference to the origin of the definition for the <rap:RapSheet> element. The @version attribute must specify the version of the rap sheet definition to which the XML document conforms. For this specification, @version must have the value "3.0.0".

Example:

```
<rap:RapSheet reportedDate="2002-02-08"
  xmlns="http://www.nlets.org/rapsheet/3.0.0"
  version="3.0.0">
  ...
</rap:RapSheet>
```

The namespace in the above example is coded as a default name space. It would apply to all sub-elements within a <rap:RapSheet> instance that do not have an explicitly

cited namespace. For clarity, all elements in this specification have an explicit namespace.

### 3.1 Introduction (<rap:Introduction>)

The <rap:Introduction> element contains reference information gathered from the inquiry transaction (e.g., IQ, FQ, or CHR) and also special caveats relating to use and dissemination of the criminal history record.

**Rap Sheet Request Reference (<rap:RapSheetRequest>):** Information contained in each sub-element is derived from the request transaction to which the response corresponds. See note below for attribute @id usage.

*Note: For responses to Interstate Identification Index (III) inquiries, the information required for the <rap:RapSheetRequest> sub-elements is available in the \$A.CHR message sent to a state by the FBI. In addition to the sub-elements, it is necessary to store a unique value in the id attribute of the <rap:RapSheetRequest> element that will allow responses from multiple states to be uniquely identified with a single inquiry. States generating a rap sheet in response to a \$A.CHR message must copy the value of the /NCF field into the @id attribute.*

#### **FBI Number (<j:PersonFBIID>):**

**GJXDM:** A number issued by the FBI's Automated Fingerprint Identification System (AFIS) based on submitted fingerprints.

The element contains the subject's FBI Number. (EFTS 2.014) The FBI Number element is mandatory in the absence of a State Bureau Identification Number element. If neither FBI nor State Bureau Identification numbers are available, the <j:PersonFBIID> element and the <j:PersonStateID> should be transmitted each with a value of <j:ID>Unknown</j:ID> .

#### **State Identification Number (<j:PersonStateID>):**

**GJXDM:** A number issued by a state Automated Fingerprint Identification System (AFIS) based on submitted fingerprints. Sometimes referred to as a State ID number or a SID.

The SID Number element is mandatory in the absence of an FBI Number element; see note above. The element contains the subject's SID from request. (EFTS 2.015) The actual value of the number is reported in a <j:ID> subelement.

*Note: For responses to Interstate Identification Index (III) inquiries, the information required for the <j:PersonFBIID> and <j:PersonStateID> sub-elements is available in the \$A.CHR message sent to a state by the FBI.*

#### **Social Security Number (<j:PersonSSNID>):**

**GJXDM:** A 9-digit numeric identifier assigned to a living person by the U.S. Social Security Administration. A social security number of a person. Sometimes referred to as a SSN.

This is an optional element included for implementations of the rapsheet where Social Security Number is used as an inquiry parameter. See definition under `<j:Subject>` for coding detail. The actual value of the number is reported in a `<j:ID>` subelement.

**Drivers License Number (`<j:PersonDriverLicenseID>`):**

**GJXDM:** Information about an identifier used to refer to a specific person who has obtained a driver's license. Sometimes referred to as driver license number, dINumber.

This is an optional element included for implementations of the rapsheet where Drivers License Number is used as an inquiry parameter. See definition under `<j:PersonSubject>` for coding detail. The actual value of the number is reported in a `<j:ID>` subelement.

**Miscellaneous Number (`<j:PersonOtherID>`):**

**GJXDM:** Information about an identifier with a type that is not explicitly defined in the standard that refers to a person within a certain domain, e.g., state ID cards or other non-license and non-fingerprint based IDs.

This is an optional element included for implementations of the rapsheet where a miscellaneous number (like passport or selective service number) is used as an inquiry parameter. The `<j:PersonIDTypeCode>` and `<j:IDIssuingAuthorityText>` subelements are required. The actual value of the number is reported in a `<j:ID>` subelement. See definition under `<j:PersonSubject>` for coding detail.

**Name (`<j:PersonFullName>`):**

**GJXDM:** A complete name of a person.

This is an optional element containing the name of the subject used in the request for the criminal history record. Entries must be coded in the following order: surname, comma, space, firstname or initial, space, middle name(s) or initial(s) if any (if multiple, space separated), space, suffix if any. (EFTS 2.018).

**Sex (`<j:PersonSexText>`):**

**GJXDM:** The gender or sex of a person.

This is an optional element included for implementations of the rapsheet where Sex is used as an inquiry parameter. See definition under `<j:PersonPhysicalDetails>` for coding detail.

**Race (`<j:PersonRaceText>`):**

**GJXDM:** A classification of a person based on factors such as geographical locations and genetics.

This is an optional element included for implementations of the rapsheet where Race is used as an inquiry parameter. See definition under `<j:PersonPhysicalDetails>` for coding detail.



**Date of Birth (<j:PersonBirthDate>):**

**GJXDM:** A date a person was born.

This is an optional element included for implementations of the rapsheet where Date of Birth is used as an inquiry parameter. See definition under <j:PersonPhysicalDetails> for coding detail.

**Purpose Code (<rap:PurposeCode>):**

The element contains a code identifying the purpose for which the rap sheet will be used. Valid values are:

- A – Administrative file maintenance
- C – Criminal justice
- D – Domestic violence and stalking, civil and criminal court cases
- F – Firearms-related background checks
- H – Public housing background checks
- I – Interstate approved non-criminal justice licensing and employment background checks
- J – criminal justice employment background checks
- S – National security background checks
- V – Visa applications initiated by US Department of State
- X – Caregiver background checks, exigent circumstances

**Attention Reference (<rap:Attention>):**

This element contains identification of the person or entity to whose attention the rap sheet response will be directed.

Example:

```
<rap:RapSheetRequest id="0HJGTM1THN">
  <j:PersonFBIID>
    <j:ID>900660</j:ID>
  </j:PersonFBIID>
  <j:PersonStateID>
    <j:IDIssuingAuthorityText>CA</j:IDIssuingAuthorityText>
    <j:ID>CA99999978</j:ID>
  </j:PersonStateID>
  <rap:PurposeCode>C</rap:PurposeCode>
  <rap:Attention>AL 911137</rap:Attention>
</rap:RapSheetRequest>
```

**Record Caveat Information (<rap:Caveat>):**

The optional Record Caveat Information contains a free-text message relating to the production or use of the rapsheet. This element is *not* intended to report information about the subject person. Use <j:SubjectOffenderNoticeText> or <j:SubjectCautionInformationCaveat> under <j:Subject> to report information about the subject. The <rap:Caveat> element is a wrapper for the text, reference date, and issuing authority elements that follow.

**Caveat Literal (<j:CaveatText>):**

**GJXDM:** A warning, caution, qualifier, or explanation.

An explanation about the rapsheet.

Example:

```
<j:CaveatText>This record is provided in response to your request. Use of the information contained in this record is governed by state and federal law.</j:CaveatText>
```

**Record Caveat Reference Date (<j:CaveatReferenceDate>):**

**GJXDM:** A date a caveat was issued.

The <j:CaveatReferenceDate> element shows the date on which the caveat message was generated. The format for the date is CCYY[-MM[-DD]].

Example:

```
<j:CaveatReferenceDate>2002-02-08</j:CaveatReferenceDate>
```

**Record Caveat Issuing Authority (<j:CaveatIssuingAuthorityName>):**

**GJXDM:** A name of a person or organization which issued a caveat.

The <j:CaveatIssuingAuthorityName> element should be set to the NCIC state two-letter code from which the caveat statement was issued; a value of "Unknown" is also allowable.

Example:

```
<j:CaveatIssuingAuthorityName>CA</j:CaveatIssuingAuthorityName>
```

**Control Data (<rap:ControlData>):**

This optional element contains implementation-specific information. The @typeText attribute should be used to identify a particular usage of control data, and distinguish between multiple types of control data that might be transmitted. The @issuingAuthorityName attribute may also be used.

*Note: For responses to Interstate Identification Index (III) inquiries, it is expected that "control data" will be contained in elements outside the rapsheet -- in an envelope, or wrapper -- and that this occurrence of the element will not be used. Control data for III consists of any alphanumeric information supplied by the requester and expected to be echoed back by the responder. The <rap:ControlData> element has been included in <rap:Introduction> for implementations other than III where it is necessary to embed this kind of information in the rapsheet document itself.*

Examples:

```
<rap:ControlData typeText="Request Index Number">273850293858</rap:ControlData>
```

```
<rap:ControlData typeText="Mail Rapsheet To">  
    Crime Information Bureau  
    PO Box 2718  
    Madison, WI 53701-2718  
</rap:ControlData>
```

## 3.2 Person Identification (<j:Subject>)

**GJXDM:** A person who is involved or suspected of being involved in an incident or criminal activity. This person may be the focus of an investigation or legal process, but is not necessarily the sole or primary focus.

The Subject is the person whose criminal history rapsheet is being reported. The Person Identification element contains information about the subject's identity.

### Subject's Primary Name (<j:PersonName>):

**GJXDM:** A name by which a person is known.

This required element must include one name for the record subject. This name may be considered to be the "Primary" name, although that distinction is known sometimes to be arbitrary, or merely a matter of circumstance or convenience. See <j:PersonAlternateName> below for reporting additional names.

### Prefix (<j:PersonPrefixName>):

**GJXDM:** A title or honorific used by a person, e.g., Dr., Judge, General, Ms.

This optional element contains an abbreviation or title appearing before the record subject's name. Examples include Dr, Rev, Hon, or Mrs. These should appear without punctuation.

### First (<j:PersonGivenName>):

**GJXDM:** A first name of a person.

This optional element contains the record subject's first or given name.

### Middle (<j:PersonMiddleName>):

**GJXDM:** A middle name of a person.

This optional element contains the record subject's middle name.

### Last (<j:PersonSurName>):

**GJXDM:** A last name or family name of a person.

This element contains the record subject's last, surname, or family name. The last name must be supplied but the remaining elements are all optional.

### Suffix (<j:PersonSuffixName>):

**GJXDM:** A component that is appended after the family name that distinguishes members of a family with the same given, middle, and last name, e.g., Jr, Sr, III, or otherwise qualifies the name (e.g., MD, LLD, PhD).

This optional element contains additional components of the record subject's name and generational suffixes, for example, Jr, Sr, III, etc., corresponding to the record subject. These should appear without punctuation.

Example:

```
<j:PersonName>
  <j:PersonPrefixName>Mr</j:PersonPrefixName>
  <j:PersonGivenName>John</j:PersonGivenName>
  <j:PersonMiddleName>P</j:PersonMiddleName>
  <j:PersonSurName>Jones</j:PersonSurName>
  <j:PersonSuffixName>Jr</j:PersonSuffixName>
</j:PersonName>
```

### **Subject's Alias Name(s) (<j:PersonAlternateName>):**

**GJXDM:** An alternate name used by a person. Sometimes referred to as an AKA.

This optional element may be repeated as many times as necessary to report all additional names for the record subject, including "also known as" (AKA) alias names. The detail elements below have the same definition and usage as for <j:PersonName>.

**Prefix (<j:PersonPrefixName>):**

**First (<j:PersonGivenName>):**

**Middle (<j:PersonMiddleName>):**

**Last (<j:PersonSurName>):**

**Suffix (<j:PersonSuffixName>):**

Example:

```
<j:PersonAlternateName>
  <j:PersonGivenName>Bill</j:PersonGivenName>
  <j:PersonSurName>Williams</j:PersonSurName>
</j:PersonAlternateName>
```

### **Residence (<j:Residence>):**

**GJXDM:** Details about the place in which a person lives.

As much information as available should be entered concerning the subject's residence location. All subelements below pertain to the physical location where the subject resided at the time the information was provided. The <j:Residence> element may contain the j:@reportedDate attribute showing the date on which the residence was reported. The format for the date is CCYY[-MM[-DD]].

**Physical Location (<j:LocationAddress>):**

**GJXDM:** Details about an address for a location, e.g., a postal address.

**Address Text (<j:AddressFullText>):**

**GJXDM:** A complete address.

This element may be used when the address components cannot be broken down into the GJXDM components. This element simply contains an unformatted text string containing the location's address.

### **Street information (<j:LocationStreet>)**

**GJXDM:** Details about a street.

#### **Street information (<j:StreetFullText>):**

**GJXDM:** A complete street reference, e.g., "123 Main Street NW".

This element would contain an entire address line, like 1565 N. Park Place. This element may be repeated if necessary.

### **Apartment or Suite information**

#### **(<j:LocationSecondaryUnitText>):**

**GJXDM:** A piece of information used to identify a particular unit within a specific location. Examples of this could be an apartment number or a suite number.

This element holds an optional apartment or a suite number for this location.

### **Post Office Box information**

#### **(<j:AddressMailDeliveryUnitText>):**

**GJXDM:** A complete Post Office box number or other mail delivery point for a location, e.g., "PO BOX 12345"

This element holds an optional post office box number for this location.

### **City (<j:LocationCityName>):**

**GJXDM:** A name of a city or town.

This element contains the city for this location.

### **County (<j:LocationCountyName>):**

**GJXDM:** A code identifying a county, parish, or vicinage.

This element contains the county for this location.

### **State (<j:LocationStateName>):**

**GJXDM:** A name of a state, commonwealth, province, or other subregion of a country.

This element contains the name of the state.

### **Zip code (<j:LocationPostalCodeID>):**

**GJXDM:** A zip code or postal code.

This element contains the 5 or 9 digit zipcode or foreign postal code for this location. The actual value of the number is reported in a <j:ID> subelement.

### **Country (<j:LocationCountryName>):**

**GJXDM:** A name of a country.

This element contains the country for this location.

### **Residence Phone (<j:LocationContactInformation>):**

**GJXDM:** Details about contact information for a location.

This element contains the telephone number of the subject at this residence.

**Residence Phone (<j:LocationContactInformation>):**

**GJXDM:** A telephone number of a person or organization.

**Phone Number (<j:TelephoneNumberFullID>):**

**GJXDM:** A full length telephone identifier.

This element contains the full phone number.

Examples:

```
<j:Residence reportedDate="1987-09-15">
  <j:LocationAddress>
    <j:AddressFullText>
      1565 N Park Place, Hamilton, NJ 08610
    </j:AddressFullText>
  </j:LocationAddress>
</j:Residence>

<j:Residence reportedDate="1989-04-22">
  <j:LocationAddress>
    <j:LocationStreet>
      <j:StreetFullText>1565 N Park Place</j:StreetFullText>
    </j:LocationStreet>
    <j:LocationSecondaryUnitText>Apt
B</j:LocationSecondaryUnitText>
    <j:LocationCityName>Hamilton</j:LocationCityName>
    <j:LocationStateName>NJ</j:LocationStateName>
    <j:LocationPostalCodeID>
      <j:ID>08610-1234</j:ID>
    </j:LocationPostalCodeID>
  </j:LocationAddress>
  <j:LocationContactInformation>
    <j:ContactTelephoneNumber>
      <j:TelephoneNumberFullID>608-266-
0872</j:TelephoneNumberFullID>
    </j:ContactTelephoneNumber>
  </j:LocationContactInformation>
</j:Residence>
```

**Employment (<j:Employment>):**

**GJXDM:** Details about the employment of a person.

This element reports information about the subject's employment.

Multiple occurrences may appear. The <j:Employment> element may contain the reportedDate attribute showing the date on which the employment was reported. The format for the date is CCYY[-MM[-DD]]. Either the <j:EmploymentOccupationText> or <j:EmploymentEmployerName> or both subelements must appear.

**Employer (<j:EmploymentEmployerName>):**

**GJXDM:** A name of an employer.

This could be an organization name, a person's name, unemployed, or self employed. This element contains the record subject's employer name.

**Occupation (<j:EmploymentOccupationText>):**

**GJXDM:** A specific type of employment or occupation.

This element contains the record subject's occupation. Occupation may be reported as "Student" if the record subject is a full-time student.

**Employment Address (<j:EmploymentLocation>):**

**GJXDM:** A location where a person works.

As much information as available should be entered concerning the employer's location and means of contact. All subelements below pertain to the physical location where the subject was employed.

**Physical Location (<j:LocationAddress>):**

**GJXDM:** Details about an address for a location, e.g., a postal address.

**Address Text (<j:AddressFullText>):**

**GJXDM:** A complete address.

This element may be used when the address components cannot be broken down into the GJXDM components. This element simply contains an unformatted text string containing the location's address.

**Street information (<j:LocationStreet>)**

**GJXDM:** Details about a street.

**Street information (<j:StreetFullText>):**

**GJXDM:** A complete street reference, e.g., "123 Main Street NW".

This element would contain an entire address line, like 1565 N. Park Place. This element may be repeated if necessary.

**Apartment or Suite information**

**(<j:LocationSecondaryUnitText>):**

**GJXDM:** A piece of information used to identify a particular unit within a specific location. Examples of this could be an apartment number or a suite number.

This element holds an optional apartment or a suite number for this location.

**Post Office Box information**

**(<j:AddressMailDeliveryUnitText>):**

**GJXDM:** A complete Post Office box number or other mail delivery point for a location, e.g., "PO BOX 12345"

This element holds an optional post office box number for this location.

**City (<j:LocationCityName>):**

**GJXDM:** A name of a city or town.

This element contains the city for this location.

**County (<j:LocationCountyName>):**

**GJXDM:** A code identifying a county, parish, or vicinage.

This element contains the county for this location.

**State (<j:LocationStateName>):**

**GJXDM:** A name of a state, commonwealth, province, or other subregion of a country.

This element contains the name of the state.

**Zip code (<j:LocationPostalCodeID>):**

**GJXDM:** A zip code or postal code.

This element contains the 5 or 9 digit zipcode or foreign postal code for this location. The actual value of the number is reported in a <j:ID> subelement.

**Country (<j:LocationCountryName>):**

**GJXDM:** A name of a country.

This element contains the country for this location.

**Employer's Phone (<j:LocationContactInformation>):**

**GJXDM:** Details about contact information for a location.

This element contains the telephone number of the subject's employer.

**Residence Phone (<j:LocationContactInformation>):**

**GJXDM:** A telephone number of a person or organization.

**Phone Number (<j:TelephoneNumberFullID>):**

**GJXDM:** A full length telephone identifier.

This element contains the full phone number.

Example:



```

<j:Employment reportedDate="1987-09-15">
  <j:EmploymentEmployerName>Carful
  Cabs</j:EmploymentEmployerName>
  <j:EmploymentOccupationText>Taxi
  Driver</j:EmploymentOccupationText>
  <j:LocationAddress>
    <j:AddressFullText>
      1565 N Park Place, Hamilton, NJ 08610
    </j:AddressFullText>
  </j:LocationAddress>
  <j:LocationContactInformation>
    <j:ContactTelephoneNumber>
      <j:TelephoneNumberFullID>608-242-
2000</j:TelephoneNumberFullID>
    </j:ContactTelephoneNumber>
  </j:LocationContactInformation>
</j:Employment>

```

### **Date of Birth (<j:PersonBirthDate>):**

**GJXDM:** A date a person was born.

Each PersonBirthDate element contains a date of birth associated with the record subject. Report all known dates of birth. The format for the date is CCYY[-MM[-DD]]. (EFTS 2.022)

Example:

```
<j:PersonBirthDate>1953-02-12</j:PersonBirthDate>
```

### **Place of Birth (<j:PersonBirthLocation>):**

**GJXDM:** Details about a place where a person was born.

The place of birth. (EFTS 2.020)

#### **Birth State or Country (<j:LocationName>):**

**GJXDM:** A name of a location.

The place of birth must be represented in the <j:LocationName> element by the name of a state in or territorial possession of the U.S., the name of a state in Mexico, the name of a Canadian province, or the name of the foreign country where the person was born.

Examples:

```

<j:PersonBirthLocation>
  <j:LocationName>Virginia</j:LocationName>
</j:PersonBirthLocation>

```

```
<j:PersonBirthLocation>
  <j:LocationName>Unknown</j:LocationName>
</j:PersonBirthLocation>
```

**Date of Death (<j:PersonDeathDate>):**

**GJXDM:** A date a person died or was declared legally dead.

The <j:PersonDeathDate> element may contain the date of a subject’s death. The format for the date is CCYY[-MM[-DD]]. This element should be omitted if the subject person has not been reported to be deceased.

Example:

```
<j:PersonDeathDate>2002-05-12</j:PersonDeathDate>
```

**Reported Deceased (<j:PersonLivingIndicator>):**

**GJXDM:** True if a person is alive, false if a person is dead.

This element should be included in the rapsheet with a value of ‘false’ if the subject person has been reported to be deceased. This element should be omitted if the subject person has not been reported to be deceased.

The j:@commentText attribute must contain information about the nature of the report. If the report was accompanied by fingerprints positively identifying the subject as deceased, use type "Fingerprint Supported". For other kinds of reports, use type "Not Fingerprint Supported". Use "Unknown" if no information is available on the type of report.

<i>Deceased Report Type</i>	
Fingerprint Supported	Not Fingerprint Supported
Unknown	

The <j:PersonLivingIndicator> element may contain the j:@reportedDate attribute showing the date of the report. The format for the date is CCYY[-MM[-DD]]. The j:@reportingOrganizationText attribute may be included to indicate an organization (name or id) that provided the information.

Example:

```
<j:PersonLivingIndicator
  reportingOrganizationText="WI041015Y"
  reportedDate="2001-02-27"
  commentText="Fingerprint
Supported">false</j:PersonLivingIndicator>
```

**Sex Offender Registration Status (<j:PersonRegisteredOffenderIndicator>):**

**GJXDM:** True if a person is required to register as an offender; false otherwise.

Report “true” if a person is listed on a state’s sex offender registry, otherwise this element should be omitted. The j:@reportingOrganizationText attribute may be included to indicate the state of registration.

Example:

```
<j:PersonRegisteredOffenderIndicator
  reportingOrganizationText="Arkansas">true
</j:PersonRegisteredOffenderIndicator>
```

**IFFS Disqualified (<j:PersonFirearmSalesDisqualifiedIndicator>):**

**GJXDM:** True if a person is prohibited from purchasing firearms; false otherwise.

This element reports the III flag for Firearm Sales.

Prior to May 13, 2001, this was the Felon Identification in Firearms Sales (FIFS) flag. Flag values were: F – used to indicate a subject’s record contains at least one felony conviction; M – used to show a subject’s record contains only misdemeanor convictions and no court action is pending; and X – used to show the status of the record is unknown or that court action is pending.

On May 13, 2001, this information became known as the Identification for Firearm Sales Disqualifier. Acceptable values are: D – Disqualified, C – Cleared, and X – used to show the status of the record is unknown or that court action is pending.

The <j:PersonFirearmSalesDisqualifiedIndicator> element only accepts values of “true” or “false.” Use the table below to convert from FIFS/IFFS values to true or false. Report the FIFS/IFFS values in the j:@commentText attribute according to the table.

The j:@reportingOrganizationText attribute may be included to indicate an organization (name or id) that provided the information.

<i>Firearm Sales Indicator</i>		
FIFS/IFFS value	Rapsheet element value	Rapsheet @commentText
F – felon	true	Felon
M – misdemeanor	false	Misdemeanor Only
D – disqualified	true	Disqualified
C - cleared	false	Cleared
X –	false	Unknown or Pending

unknown/pending		
-----------------	--	--

Example:

```
<j:PersonFirearmSalesDisqualifiedIndicator
  reportingOrganization="Minnesota"
  commentText="Disqualified">true
</j:PersonFirearmSalesDisqualifiedIndicator>
```

### **Person Identification Numbers (<j:PersonAssignedIDDetails>):**

**GJXDM:** A set of details about identifications issued to a person.

This group element contains ID numbers.

#### **Social Security Number (<j:PersonSSNID>):**

**GJXDM:** A 9-digit numeric identifier assigned to a living person by the U.S. Social Security Administration.

A social security number of a person. Sometimes referred to as a SSN. This element contains the subject's social security number. This number will be entered as nine consecutive numeric characters with no embedded punctuation or special characters. (EFTS 2.016)

Example:

```
<j:PersonSSNID>
  <j:ID>220565860</j:ID>
</j:PersonSSNID>
```

#### **Driver's License Number (<j:PersonDriverLicenseID>):**

**GJXDM:** Information about an identifier used to refer to a specific person who has obtained a driver's license.

Sometimes referred to as driver license number, dlNumber. The element contains a subject's driver's license number for a given state. The NCIC state two-letter code must be specified in the <j:IDIssuingAuthorityText> subelement; "Unknown" is also an allowable value.

Example:

```
<j:PersonDriverLicenseID>
  <j:IDIssuingAuthorityText>WI</j:IDIssuingAuthorityText>
  <j:ID>C4556289248R</j:ID>
</j:PersonDriverLicenseID>
```

#### **FBI Number (<j:PersonFBIID>):**

**GJXDM:** A number issued by the FBI's Automated Fingerprint Identification System (AFIS) based on submitted fingerprints.

The FBI Number element is mandatory without a SID Number element (<j:PersonStateID>). The value of the element contains the subject's FBI

Number. If neither FBI nor State ID numbers are known, both may be reported as "Unknown". (EFTS 2.014)

Example:

```
<j:PersonFBIID>
  <j:ID>62660CA12</j:ID>
</j:PersonFBIID>
```

### **State Identification Number (<j:PersonStateID>):**

**GJXDM:** A number issued by a state Automated Fingerprint Identification System (AFIS) based on submitted fingerprints.

Sometimes referred to as a State ID number or a SID. The SID Number field is mandatory without an FBI Number field (<j:PersonFBIID>). The element contains the subject's SID Number assigned by the responding state(s). Although a SID number often contains a state code, the state code must also be specified in the <j:IDIssuingAuthorityText> subelement.(EFTS 2.015)

Example:

```
<j:PersonStateID>
  <j:IDIssuingAuthorityText>CA</j:IDIssuingAuthorityText>
  <j:ID>CA9936278</j:ID>
</j:PersonStateID>
```

### **Correctional Number (<j:PersonOtherID>):**

**GJXDM:** Information about an identifier with a type that is not explicitly defined in the standard that refers to a person within a certain domain, e.g., state ID cards or other non-license and non-fingerprint based IDs.

This optional element contains a correctional subject's identification number for a given state. The NCIC state two-letter code must be specified in the IDIssuingAuthorityText element; "Unknown" is also an allowable value.

Example:

```
<j:PersonOtherID>
  <j:IDTypeText>CorrectionalID</j:IDTypeText>
  <j:IDIssuingAuthorityText>IL</j:IDIssuingAuthorityText>
  <j:ID>742786</j:ID>
</j:PersonOtherID>
```

### **Miscellaneous ID Number (<j:PersonOtherID>):**

**GJXDM:** Information about an identifier with a type that is not explicitly defined in the standard that refers to a person within a certain domain, e.g., state ID cards or other non-license and non-fingerprint based IDs.

The element contains other identifying numbers issued to the subject (EFTS 2.909). The NCIC state or country two-letter code, or other locality descriptor must be specified in the <j:IDIssuingAuthorityText> subelement. The subelement <j:IDTypeText> name is also required; valid values for the IDTypeText name are

<i>PersonIDNumber Types</i>	
Air Force Serial	Navy Serial
Air National Guard Serial	Offender ID
Alien Registration	Originating Agency Police ID
Army Serial	Passport
Bureau Fugitive Index	Personal Identification
Canadian Social Insurance	Port Security Card
Correctional ID	Royal Canadian Mounted Police ID
Identification Order	Selective Service
Marine Corps Serial	State ID Card
Mariner's Document ID	US Coast Guard Serial
National Agency Case	Veterans Administration Claim
National Guard Serial	Unknown

Both the NCIC codes and EFTS codes use the code "AS" for Army, Air National Guard, and National Guard Serial numbers. If unable to distinguish a specific usage from existing data, map the code "AS" to "Army Serial."

Example:

```
<j:PersonOtherID>
  <j:IDTypeText>State ID Card</j:IDTypeText>
  <j:IDIssuingAuthorityText>WI</j:IDIssuingAuthorityText>
  <j:ID>C4556289248R</j:ID>
</j:PersonOtherID>
```

### **Person Physical Description (<j:PersonPhysicalDetails>):**

**GJXDM:** A set of details about the physical appearance of a person.

This group element contains descriptive information about the record subject.

#### **Height (<j:PersonHeightMeasure>):**

**GJXDM:** A measurement of the height of a person.

This element contains the record subject's height. (EFTS 2.913)

The <j:PersonHeightMeasure> element must contain the j:@personHeightUnitCode attribute with a value of "ncic".

**GJXDM:** An NCIC formatted three-digit person height field. The first digit represents the height of the person in feet, the second two digits represent the remainder of the height in inches, e.g. six feet two inches is represented as "602".

When reported in feet and inches, the first (leftmost) digit is used to show feet while the two rightmost characters are used to show the inches between 00 and 11.

The <j:PersonHeightMeasure> element may contain the j:@reportedDate attribute containing the date on which the associated height was reported. The format for the date is CCYY[-MM[-DD]].

Example:

```
<j:PersonHeightMeasure personHeightUnitCode="ncic"
reportedDate="1994-07-04">601</j:PersonHeightMeasure>
```

**Weight (<j:PersonWeightMeasure>):**

**GJXDM:** A measurement of the weight of a person.

This element contains the record subject's weight in pounds as a three-character numeric. If weight is unknown, enter 000. (EFTS 2.914)

The <j:PersonWeightMeasure> element must contain the j:@personWeightUnitCode attribute with a value of "ncic".

**GJXDM:** An NCIC formatted three-digit person weight field, where the three digits represents the weight of the person in pounds

The <j:PersonWeightMeasure> element may contain the j:@reportedDate attribute containing the date on which the associated weight was reported. The format for the date is CCYY[-MM[-DD]].

Example:

```
<j:PersonWeightMeasure personWeightUnitCode="ncic"
reportedDate="1997-03-21">180</j:PersonWeightMeasure>
```

**Eye Color (<j:PersonEyeColorText>):**

**GJXDM:** The color of a person's eyes.

This element contains the subject's eye color. Use an eye color value from the following table. (EFTS 2.915)

<i>Eye Color</i>	
Black	Hazel
Blue	Maroon
Brown	Multicolored
Gray	Pink
Green	Unknown

The <j:PersonEyeColorText> element may contain the j:@reportedDate attribute containing the date on which the associated eye color was reported. The format for the date is CCYY[-MM[-DD]].

Example:

```
<j:PersonEyeColorText reportedDate="1988-03-21">
  Blue</j:PersonEyeColorText>
```

**Hair Color (<j:PersonHairColorText>):**

**GJXDM:** The color of a person's hair.

This element contains the subject's hair color. (EFTS 2.916) Use a hair color from the following table.

<i>Hair Color</i>	
Bald	Orange
Black	Pink
Blonde Or Strawberry	Purple
Blue	Red Or Auburn
Brown	Sandy
Gray Or Partially Gray	White
Green	Unknown

The <j:PersonHairColorText> element may contain the j:@reportedDate attribute containing the date on which the associated hair color was reported. The format for the date is CCYY[-MM[-DD]].

The ANSI-NIST-FBI Electronic Fingerprint Transmission Specification and the NCIC Code Table have fourteen codes for the HAI field code. Systems using the ANSI-NIST or NCIC values should translate according to the following table:

<i>Translation of Hair Color Codes</i>			
Reported Hair Color	ANSI-NIST Code	NCIC Code	Rapsheet Value
Bald	BAL	BLD	Bald
Black	BLK	BLK	Black
Blonde or Strawberry	BLN	BLN	Blonde Or Strawberry
Brown	BRO	BRO	Brown
Gray or Partially Gray	GRY	GRY	Gray Or Partially Gray
Red or Auburn	RED	RED	Red Or Auburn
Sandy	SDY	SDY	Sandy
White	WHI	WHI	White



<i>Translation of Hair Color Codes</i>			
Reported Hair Color	ANSI-NIST Code	NCIC Code	Rapsheet Value
Blue	BLU	BLU	Blue
Green	GRN	GRN	Green
Orange	ONG	ONG	Orange
Pink	PNK	PNK	Pink
Purple	PLE	PLE	Purple
Unknown	XXX	XXX	Unknown

Example:

```
<j:PersonHairColorText reportedDate="1989-07-04">
  Brown</j:PersonHairColorText>
```

**Sex (<j:PersonSexText>):**

**GJXDM:** The gender or sex of a person.

This element contains the record subject's sex. (EFTS 2.912) Valid values are: Male; Female; Other; Unknown.

The ANSI-NIST-FBI Electronic Fingerprint Transmission Specification has seven codes for element SEX. Systems using the ANSI-NIST values should translate according to the following table:

<i>Translation of Sex Codes</i>		
Reported Sex	ANSI-NIST Code	Rapsheet Value
Female	F	Female
Male Impersonator	G	Other
Male	M	Male
Female Impersonator or Transvestite	N	Other
Unreported sex; male name	Y	Unknown
Unreported sex; female name	Z	Unknown
Unknown	X	Unknown

Example:

```
<j:PersonSexText>Female</j:PersonSexText>
```

**Race (<j:PersonRaceText>):**

**GJXDM:** A classification of a person based on factors such as geographical locations and genetics.

This element reports the record subject's race. Use the predominant race code from the following table. (Categories are the same as EFTS 2.025, code values are different.)

<i>If Subject is</i>	<i>Enter Code</i>
Chinese, Japanese, Filipino, Korean, Polynesian, Indian, Indonesian, Asian Indian, Samoan, or any other Pacific Islander	Asian
A person having origins in any of the racial groups of Africa	Black
American Indian, Eskimo, or Alaskan native, or a person having origins in any of the 48 contiguous states of the United States or Alaska who maintains cultural identification through tribal affiliation or community recognition	Native American
Of indeterminate race	Unknown
Caucasian, Mexican, Puerto Rican, Cuban, Central or South American, or other Spanish culture or origin, regardless of race	White

Example:

```
<j:PersonRaceText>White</j:PersonRaceText>
```

**Skin Tone (<j:PersonSkinToneText>):**

**GJXDM:** A color or tone of a person's skin.

This element contains the subject's skin tone (complexion). Use a skin tone from the following table. (An NCIC field, but not EFTS; it is likely that many criminal history systems do not contain this data element and it should simply be omitted).

<i>Skin Tone</i>	
Albino	Medium
Black	Medium Brown
Dark	Olive
Dark Brown	Ruddy
Fair	Sallow
Light	Yellow

<i>Skin Tone</i>	
Light Brown	Unknown

Example:

```
<j:PersonSkinToneText>Albino</j:PersonSkinToneText>
```

### **Scars, Marks, and Tattoos (<j:PersonPhysicalFeature>):**

**GJXDM:** Details about a physical feature of a person. Includes scars, marks, and tattoos.

This group element will provide a literal description of a scar, mark, or tattoo. It may provide an NCIC code. (EFTS 2.921) Images may be included or referenced.

The <j:PersonPhysicalFeature> element may contain the j:@reportedDate attribute showing the date on which the scar, mark or tattoo information was obtained. The format for the date is CCYY[-MM[-DD]].

The j:@reportingOrganizationText attribute may be included to report an agency having more specific information associated with the scar, mark or tattoo. Include either agency name, ORI, or both.

#### **Scar, Mark, or Tattoo (<j:PhysicalFeatureCategoryText>)**

**GJXDM:** A general classification or categorization of a physical feature, e.g.,scar, mark, tattoo, missing limb.

Rapsheet values are, “Scar” “Mark” or “Tattoo”.

#### **SMT Code (<j:PhysicalFeatureTypeCode>)**

**GJXDM:** A code identifying a type of physical feature.

One of <j:PhysicalFeatureTypeCode> or <j:PhysicalFeatureDescriptionText> is required. The type code is an element for reporting NCIC SMT codes. (EFTS 2.026)

#### **Description (<j:PhysicalFeatureDescriptionText>)**

**GJXDM:** A description of a physical feature.

One of <j:PhysicalFeatureTypeCode> or <j:PhysicalFeatureDescriptionText> is required. The description text is literal description of the scar, mark, or tattoo.

#### **Scar, Mark, or Tattoo Photo (<j:PhysicalFeatureImage>)**

**GJXDM:** A digital image of a physical feature, e.g., tattoo.

This optional element contains an image of a scar, mark, or tattoo; or, if only the `j:BinaryCaptureOrganization` is reported, then the agency must be contacted to obtain an image on file. This element may occur multiple times. Only one image is to be coded per `<j:PhysicalFeatureImage>` element.

### **Image (<j:BinaryObject.Base64)**

**GJXDM:** A binary encoding of data, e.g., a binary encoding of a picture, photo, image, graphic, sound, or video.

Optional. If the actual image is contained here, it must be encoded using the Base64 algorithm which transforms binary data into text characters.

### **Image Format (<j:BinaryFormatText)**

**GJXDM:** A format of a binary object, e.g., “image/jpeg”.

The `<j:BinaryFormatText>` is an optional element containing the format in which the SMT image is stored. Values include “image/gif” “image/tiff” “image/jpeg” “application/postscript” “video/mpeg”. Use “image/wsqa” for fingerprint images compressed using the ANSI/NIST/FBI recommended algorithm wavelet scalar quantization. The format is the binary format of the image prior to Base64 encoding, or the format of the image that is the target of a hyperlink (see below).

### **Image Size (<j:BinarySizeValue)**

**GJXDM:** A size of a binary object in kilobytes.

The `<j:BinarySizeValue>` is an optional element containing the approximate size of the original image (before encoding).

### **Image Type (<j:BinaryTypeText)**

**GJXDM:** A type of binary object that is encoded, e.g., mug shot, driver license picture, audio confession..

The `<j:BinaryTypeText>` element is required and must contain information about the type of photo specified in the image. It may contain the value “Scar” “Mark” “Tattoo” or other type appropriate to the image.

### **Image Description or Comment(<j:BinaryDescriptionText)**

**GJXDM:** A textual description of a binary object.

An optional element for reporting additional free-text information about the photo. This may be a note or comment about the photo.

### Image Hyperlink (<j:BinaryReferenceID)

**GJXDM:** Identifies a url or file reference of the binary object.

An optional element for reporting a hyperlink to the photo. The hyperlink may be a URL or URI or description of a file location. The actual text value of the hyperlink is reported in an <j:ID> subelement.

Example:

```
<j:BinaryReferenceID>
  <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
</j:BinaryReferenceID>
```

### Date of image (<j:BinaryCaptureDate)

**GJXDM:** A date on which the data represented by the binary object is captured, e.g., an image taken or an audio recorded..

This element may contain the date on which the scar, mark or tattoo image was obtained. The format for the date is CCYY[-MM[-DD]].

### Image Agency (<j:BinaryCaptureOrganization)

**GJXDM:** An organization which captured or created a binary.

This is the only required element for an image. Use OrganizationName of “Unknown” if no agency is known. If the agency is reported without image or hyperlink data, then the agency has an image on file and can be contacted to obtain one. Use subelements <j:OrganizationName> and/or <j:OrganizationORIID><ID>.

Example:

```
<j:PersonPhysicalFeature reportedDate="1999-06-04">
  <j:PhysicalFeatureCategoryText>
    tattoo</j:PhysicalFeatureCategoryText>
  <j:PhysicalFeatureTypeCode>TAT
  FARM</j:PhysicalFeatureTypeCode>
  <j:PhysicalFeatureDescriptionText>
    Dragon tattoo on right forearm.
  </j:PhysicalFeatureDescriptionText>
  <j:PhysicalFeatureImage>
    <j:BinaryObject.Base64>
      dkriJEORjSo9rj6+/kDIsl...DKRHkdiopq
    </j:BinaryObject.Base64>
    <j:BinaryFormatText>image/jpeg</j:BinaryFormatText>
    <j:BinarySizeValue>55</j:BinarySizeValue>
```

```

    <j:BinaryTypeText>Tattoo</j:BinaryTypeText>
    <j:BinaryDescriptionText>Color
pic</j:BinaryDescriptionText>
    <j:BinaryReferenceID>
      <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
    </j:BinaryReferenceID>
    <j:BinaryCaptureDate>1999-06-04</j:BinaryCaptureDate>
    <j:BinaryCaptureOrganization>
      <j:OrganizationName>CIB</j:OrganizationName>
      <j:OrganizationORIID>
        <j:ID>WI013415Y</j:ID>
      </j:OrganizationORIID>
    </j:BinaryCaptureOrganization>
  </j:PhysicalFeatureImage>
</j:PersonPhysicalFeature>

```

**Person Social Description (<j:PersonSocialDetails>):**

**GJXDM:** A set of details about the social characteristics of a person.

This group element contains descriptive information about the record subject.

**Ethnicity (<j:PersonEthnicityText>):**

**GJXDM:** A person's cultural lineage.

An optional element for additional description of the subject's race, appearance, or heritage. It is the intent of this standard to comply with OMB Directive 15 (<http://www.whitehouse.gov/omb/fedreg/ombdir15.html>) minimum standards, but also to define a set of values that will be of descriptive value to the criminal justice community. Use an ethnicity value from the following table:

<i>Ethnicity</i>	
African	Mexican
Alaska Native	Middle Eastern
American Indian	Mixed
Asian Indian	Native Hawaiian
Black Or African American	Not Hispanic Or Latino
Cambodian	Pacific Islander
Central American	Pakastani
Chinese	Polynesian
European	Puerto Rican
Filipino	Samoan

<i>Ethnicity</i>	
Great Britain	Slavic
Haitian	South American
Hispanic Or Latino	Spanish Origin
Indonesian	Thai
Japanese	Unknown
Korean	Vietnamese
Malaysian	White American

Example:

```
<j:PersonEthnicityText>Hispanic Or
Latino</j:PersonEthnicityText>
```

### **Country of Citizenship (<j:PersonCitizenshipText>):**

**GJXDM:** A country that assigns rights, duties, and privileges to a person due to the person's birth or naturalization in that country.

This element reports the name of the country of which the subject is a citizen. (EFTS 2.911) This element may contain the `j:@reportedDate` attribute containing the date on which the associated data was reported. The format for the date is CCYY[-MM[-DD]].

Example:

```
<j:PersonCitizenshipText reportedDate="2001-11-29">United
States
</j:PersonCitizenshipText>
```

### **Marital Status (<j:PersonMaritalStatusText>):**

**GJXDM:** A marital status of a person, e.g., married, divorced, single, separated.

An optional element containing information about the subject's marital status. This element may contain the `j:@reportedDate` attribute containing the date on which the associated data was reported. The format for the date is CCYY[-MM[-DD]].

Use a marital status value from the following table:

<i>Marital Status</i>	
Married	Never Married
Widowed	Unmarried Partner
Divorced	Unknown

Separated	
-----------	--

Example:

```
<j:PersonMaritalStatusText reportedDate="2001-11-29">Never
Married</j:PersonMaritalStatusText>
```

### **Religion (<j:PersonReligionText>):**

**GJXDM:** A religion to which a person subscribes or believes. Describes the overall category of a person's spiritual beliefs.

This is an optional element containing information about the subject's religion. This element may contain the `j:@reportedDate` attribute containing the date on which the associated data was reported. The format for the date is CCYY[-MM[-DD]].

Example:

```
<j:PersonReligionText reportedDate="2001-11-
29">Lutheran</j:PersonReligionText>
```

### **Person Biometric Information and Photos (<j:PersonBiometricDetails>):**

**GJXDM:** A set of details about measurable biological or behavioral characteristics, which can reliably recognize the identity, or verify the claimed identity, of a person.

This group element contains information about biometric characteristics and photos for the record subject.

#### **Photo Image(s) (<j:PersonDigitalImage>):**

**GJXDM:** A photograph or image of a person in a digital format.

Entries should be made in this element if photo images are being transmitted as part of the record, if links are specified to available images, or if photos are available from the referenced agency. This element may occur multiple times. Only one image is to be coded per `<j:PersonDigitalImage>` element.

Do *not* include these types of photos here: (1)Photos of scars, marks, and tattoos should be transmitted as part of the `<j:PersonPhysicalFeature>` element; (2) Fingerprint and palm print images should be transmitted using the respective elements `<j:PersonFingerPrintSet>` and `<j:PersonPalmPrint>`.

Use the `<j:BinaryCaptureDate>` element (below) to report the date on which the image was obtained.

#### **Image (<j:BinaryObject.Base64>)**

**GJXDM:** A binary encoding of data, e.g., a binary encoding of a picture, photo, image, graphic, sound, or video.



Optional. If the actual image is contained here, it must be encoded using the Base64 algorithm which transforms binary data into text characters.

### **Image Format (<j:BinaryFormatText)**

**GJXDM:** A format of a binary object, e.g., “image/jpeg”.

The <j:BinaryFormatText> is an optional element containing the format in which the SMT image is stored. Values include “image/gif” “image/tiff” “image/jpeg” “application/postscript” “video/mpeg”. Use “image/wsqa” for fingerprint images compressed using the ANSI/NIST/FBI recommended algorithm wavelet scalar quantization. The format is the binary format of the image prior to Base64 encoding, or the format of the image that is the target of a hyperlink (see below).

### **Image Size (<j:BinarySizeValue)**

**GJXDM:** A size of a binary object in kilobytes.

The <j:BinarySizeValue> is an optional element containing the approximate size of the original image (before encoding).

### **Image Type (<j:BinaryTypeText)**

**GJXDM:** A type of binary object that is encoded, e.g., mug shot, driver license picture, audio confession..

The <j:BinaryTypeText> element is required and must contain information about the type of photo specified in the image.

Use a value from the following table. Use <j:BinaryTypeText>Mugshot for full-front facial photos. Mugshots that comply with the NIST Best Practice for the Capture of Mugshots (<http://www.itl.nist.gov/iaui/894.03/face/face.html>) are encouraged. A <j:BinaryTypeText>Identification photo is a photo of the subject other than a mugshot.

<i>Photo Type</i>	
Mugshot	Identification
Signature	Other
Unknown	

### **Image Description or Comment (<j:BinaryDescriptionText)**

**GJXDM:** A textual description of a binary object.

An optional element for reporting additional free-text information about the photo. This may be a note or comment about the photo.

### **Image Hyperlink (<j:BinaryReferenceID)**

**GJXDM:** Identifies a url or file reference of the binary object.

An optional element for reporting a hyperlink to the photo. The hyperlink may be a URL or URI or description of a file location. The actual text value of the hyperlink is reported in an <j:ID> subelement.

Example:

```
<j:BinaryReferenceID>
  <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
</j:BinaryReferenceID>
```

### **Date of image (<j:BinaryCaptureDate)**

**GJXDM:** A date on which the data represented by the binary object is captured, e.g., an image taken or an audio recorded..

This element may contain the date on which the photo image was obtained. The format for the date is CCYY[-MM[-DD]].

### **Image Agency (<j:BinaryCaptureOrganization)**

**GJXDM:** An organization which captured or created a binary.

This is the only required element for an image. Use OrganizationName of “Unknown” if no agency is known. If the agency is reported without image or hyperlink data, then the agency has an image on file and can be contacted to obtain one. Use subelements <j:OrganizationName> and/or <j:OrganizationORIID><ID>.

Examples:

```
<j:PersonDigitalImage>
  <j:BinaryObject.Base64>
    dkriJEORjSo9rj6+/kDIsl...DKRHkdiop
  </j:BinaryObject.Base64>
  <j:BinaryFormatText>image/jpeg</j:BinaryFormatText>
  <j:BinarySizeValue>15</j:BinarySizeValue>
  <j:BinaryTypeText>Identification</j:BinaryTypeText>
  <j:BinaryDescriptionText>Color
pic</j:BinaryDescriptionText>
  <j:BinaryReferenceID>
    <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
  </j:BinaryReferenceID>
  <j:BinaryCaptureDate>1999-06-04</j:BinaryCaptureDate>
  <j:BinaryCaptureOrganization>
    <j:OrganizationName>CIB</j:OrganizationName>
    <j:OrganizationORIID>
      <j:ID>WI013415Y</j:ID>
    </j:OrganizationORIID>
```

```

    </j:BinaryCaptureOrganization>
</j:PersonDigitalImage>

<j:PersonDigitalImage>
  <j:BinaryTypeText>Mugshot</j:BinaryTypeText>
  <j:BinaryCaptureOrganization>
    <j:OrganizationORIID>
      <j:ID>WI013415Y</j:ID>
    </j:OrganizationORIID>
  </j:BinaryCaptureOrganization>
</j:PersonDigitalImage>

```

**Blood Type (<j:PersonBloodTypeText>):**

**GJXDM:** A person's type of blood, e.g., A positive, O negative.

This element contains the subject's blood type. Use a value from the following table.

<i>Blood Type</i>	
A Positive	AB Positive
A Negative	AB Negative
A Unknown	AB Unknown
B Positive	O Positive
B Negative	O Negative
B Unknown	O Unknown
	Unknown

Example:

```
<j:PersonBloodTypeText>O Positive</j:PersonBloodTypeText>
```

**Fingerprint Information and Images (<j:PersonFingerprintSet>):**

**GJXDM:** A representation or an encoding of the identifying characteristics of a set of a persons fingerprints.

This group element reports the subject's fingerprint classification and may include fingerprint images or references to where images may be obtained.

**10-Print Fingerprint Classification (<j:BiometricValueText>):**

**GJXDM:** A textual representation of the value of a biometric.

The element contains the subject's full 10-print fingerprint classification by "Henry" or other method. (EFTS 2.917)

## 10-Print Fingerprint Classification Method

(**<j:BiometricEncodingMethodText>**):

**GJXDM:** A a method used to encode a biometric.

If the fingerprint classification is reported in the element above, then this element may be "Henry" or "FPC" or "Other" or "Unknown" to indicate the classification technique employed. Other classifications types may include pattern codes proprietary to a particular automated fingerprint identification system (AFIS).

**One or More Fingerprint Images (<j:Fingerprint>):**

**GJXDM:** A representation or an encoding of the identifying characteristics of a persons fingerprints.

Entries should only be made in this element if fingerprint images are being transmitted as part of the record, or if links are specified to available images. It is assumed that arresting agencies, state identification bureaus, and/or the FBI have fingerprints available. It is not necessary to transmit a full set of fingerprints. Transmitting a single right index print, or making a hyperlink available, is encouraged. This element may occur multiple times. Only one image is to be coded per <j:Fingerprint> element.

**Fingerprint Image Type (<j:FingerprintFingerText>):**

**GJXDM:** A finger of a person that has been printed or has attempted to have been printed, e.g., right thumb, left index.

Use this to name which finger image is being reported. A required subelement if <j:Fingerprint> is used. Use a value from the following table.

<i>Fingerprint Type</i>	
Right Thumb	Left Thumb
Right Index	Left Index
Right Middle	Left Middle
Right Ring	Left Ring
Right Little	Left Little
Right Hand Set	Left Hand Set
Ten Print Set	Unknown

**Fingerprint Image (<j:BiometricImage>):**

**GJXDM:** A picture of a biometric sample.

This is a group element reporting the detail of a single image

### **Image (<j:BinaryObject.Base64)**

**GJXDM:** A binary encoding of data, e.g., a binary encoding of a picture, photo, image, graphic, sound, or video.

Optional. If the actual image is contained here, it must be encoded using the Base64 algorithm which transforms binary data into text characters.

### **Image Format (<j:BinaryFormatText)**

**GJXDM:** A format of a binary object, e.g., “image/jpeg”.

The <j:BinaryFormatText> is an optional element containing the format in which the SMT image is stored. Values include “image/gif” “image/tiff” “image/jpeg” “application/postscript” “video/mpeg”. Use “image/wsqa” for fingerprint images compressed using the ANSI/NIST/FBI recommended algorithm wavelet scalar quantization. The format is the binary format of the image prior to Base64 encoding, or the format of the image that is the target of a hyperlink (see below).

### **Image Size (<j:BinarySizeValue)**

**GJXDM:** A size of a binary object in kilobytes.

The <j:BinarySizeValue> is an optional element containing the approximate size of the original image (before encoding).

### **Image Type (<j:BinaryTypeText)**

**GJXDM:** A type of binary object that is encoded, e.g., mug shot, driver license picture, audio confession..

The <j:BinaryTypeText> element is required and must contain information about the type of photo specified in the image. It is sufficient to use the value “Fingerprint” since the <j:FingerprintFingerText> element above will have information on the specific finger.

### **Image Description or Comment (<j:BinaryDescriptionText)**

**GJXDM:** A textual description of a binary object.

An optional element for reporting additional free-text information about the photo. This may be a note or comment about the photo.

### **Image Hyperlink (<j:BinaryReferenceID)**

**GJXDM:** Identifies a url or file reference of the binary object.

An optional element for reporting a hyperlink to the photo. The hyperlink may be a URL or URI or description of a file location. The actual text value of the hyperlink is reported in an <j:ID> subelement.

Example:

```
<j:BinaryReferenceID>
  <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg
</j:ID>
</j:BinaryReferenceID>
```

### Date of image (<j:BinaryCaptureDate>)

**GJXDM:** A date on which the data represented by the binary object is captured, e.g., an image taken or an audio recorded..

This element may contain the date on which the fingerprint image was obtained. The format for the date is CCYY[-MM[-DD]].

### Image Agency (<j:BinaryCaptureOrganization>)

**GJXDM:** An organization which captured or created a binary.

This is the only required element for an image. Use OrganizationName of “Unknown” if no agency is known. If the agency is reported without image or hyperlink data, then the agency has an image on file and can be contacted to obtain one. Use subelements <j:OrganizationName> and/or <j:OrganizationORIID><ID>.

Example (transmits a fingerprint image):

```
<j:PersonFingerprintSet>
  <j:Fingerprint>
    <j:FingerprintFingerText>Right
Index</j:FingerprintFingerText>
    <j:BiometricImage>
      <j:BinaryObject.Base64>
        dkriJEORjSo9rj6+/kDIsl...DKRHkdiop
      </j:BinaryObject.Base64>
      <j:BinaryFormatText>image/wsq</j:BinaryFormatText>
      <j:BinarySizeValue>75</j:BinarySizeValue>
      <j:BinaryTypeText>Fingerprint</j:BinaryTypeText>

    <j:BinaryDescriptionText>Grayscale</j:BinaryDescriptionText>
    <j:BinaryReferenceID>
      <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
    </j:BinaryReferenceID>
    <j:BinaryCaptureDate>1999-06-04</j:BinaryCaptureDate>
    <j:BinaryCaptureOrganization>
      <j:OrganizationName>CIB</j:OrganizationName>
      <j:OrganizationORIID>
```

```

        <j:ID>WI013415Y</j:ID>
        </j:OrganizationORIID>
    </j:BinaryCaptureOrganization>
</j:BiometricImage>
</j:Fingerprint>
</j:PersonFingerprintSet>

```

Example (transmits fingerprint classification):

```
<j:PersonFingerprintSet>
```

```
<j:BiometricValueText>66AA09TTPI58AA6413XI</j:BiometricValueText>
```

```
<j:BiometricEncodingMethodText>FPC</j:BiometricEncodingMethodText>
</j:PersonFingerprintSet>
```

### **DNA (<j:PersonDNA>):**

**GJXDM:** A representation or an encoding of the identifying characteristics of a person's DNA.

Entries should only be made in this element if an Agency is known to have a DNA sample of the record subject. This element allows two kinds of reporting. First, most common and useful, is to report that a DNA sample has been taken from the subject, has been coded, and is available from a specific agency. Second, not normally included in a criminal history response, is the optional ability to transmit the actual detail of the DNA code. The latter is included for those implementations that require the transmittal of the detail code.

Subelements used to report the actual DNA encoded values for this subject may be omitted. If actual codes are specified, the encoding method must be STR (short tandem repeat). The name "STR" must be specified as the value of the <j:BiometricEncodingMethodText> element if <j:DNALocus> subelements are included. The <j:PersonDNA> element may report the <j:BiometricCaptureDate> element showing the date on which the DNA sample was collected or processed, and may report the <j:BiometricCaptureOrganization> showing where DNA information is available.

#### **Encoding Method (<j:BiometricEncodingMethodText>):**

**GJXDM:** A method used to encode a biometric..

Use this element only if <j:DNALocus> element will be transmitted. The only allowable value is "STR" (short tandem repeat).

<b>Additional Comment (&lt;j:BiometricDescriptionText )</b>	<b>Description</b>	<b>or</b>
---	--------------------	-----------

**GJXDM:** A description of a biometric.

An optional element for reporting additional free-text information or comment about the DNA.

#### **Agency (<j:BiometricCaptureOrganization>):**

**GJXDM:** An organization that collected a biometric sample.

This element contains information about the agency holding or supplying DNA information. If the <j:DNALocus> elements are not transmitted, then the DNA detail information is available from the agency. One or both of the subelements <j:OrganizationName> or <j:OrganizationORIID> must be present.

**Date (<j:BiometricCaptureDate>):**

**GJXDM:** A date a biometric sample was collected..

This element contains the date the DNA sample was taken. The format for the date is CCYY[-MM[-DD]].

**DNA Detail (<j:DNALocus>):**

**GJXDM:** Location specific information regarding a person's DNA.

The STR encoded values for this subject may be transmitted using repetitive occurrences of this element. This element repeats up to 13 times for the loci used in STR (short tandem repeat) encoding. An optional 14<sup>th</sup> locus (Amelogenin) defines the subject's sex and may be included.

**DNA Detail Locus Type (<j:DNALocusTypeText>):**

**GJXDM:** The location within a strand of DNA that a value was determined..

The name of the locus must be specified. Use type values from the following table:

<i>DNA Loci Types</i>	
Amelogenin	D5S818
CSF1PO	D7S820
D13S317	D8S1179
D16S539	FGA
D18S51	TH01
D21S11	TPOX
D3S1358	vWA

**DNA Encoded Values (<j:DNALocusValue>):**

**GJXDM:** The value string for a DNA locus.

Each DNA locus may have one or two values. Repeat the <j:DNALocusValue> element for multiple values. All of the loci have numeric values except Amelogenin which uses X, Y, and XX.



## Examples:

```
<j:PersonDNA>
  <j:BiometricCaptureOrganization>
    <j:OrganizationORIID>
      <j:ID>WI041015Y</j:ID>
    </j:OrganizationORIID>
  </j:BiometricCaptureOrganization>
</j:PersonDNA>

<j:PersonDNA>

<j:BiometricEncodingMethodText>STR</j:BiometricEncodingMethodText
>
  <j:BiometricDescriptionText>
    Pursuant to sex offender registration.
  </j:BiometricDescriptionText>
  <j:BiometricCaptureOrganization>
    <j:OrganizationName>Crime Laboratory</j:OrganizationName>
    <j:OrganizationORIID>
      <j:ID>WI041015Y</j:ID>
    </j:OrganizationORIID>
  </j:BiometricCaptureOrganization>
  <j:BiometricCaptureDate>2001-02-27</j:BiometricCaptureDate>
  <j:DNALocus>
    <j:DNALocusTypeText>Amelogenin</j:DNALocusTypeText>
    <j:DNALocusValue>X</j:DNALocusValue>
    <j:DNALocusValue>Y</j:DNALocusValue>
  </j:DNALocus>
  <j:DNALocus>
    <j:DNALocusTypeText>CSF1PO</j:DNALocusTypeText>
    <j:DNALocusValue>11</j:DNALocusValue>
    <j:DNALocusValue>12</j:DNALocusValue>
  </j:DNALocus>
  <j:DNALocus>
    <j:DNALocusTypeText>D13S317</j:DNALocusTypeText>
    <j:DNALocusValue>11</j:DNALocusValue>
  </j:DNALocus>
    ...
  <j:DNALocus>
    <j:DNALocusTypeText>vWA</j:DNALocusTypeText>
    <j:DNALocusValue>14</j:DNALocusValue>
    <j:DNALocusValue>16</j:DNALocusValue>
  </j:DNALocus>
</j:PersonDNA>
```

### **Palm Print Image(s) (<j:PersonPalmPrint>):**

**GJXDM:** A representation or an encoding of the identifying characteristics of a person's palm print.

Entries should be made in this element if palm print images are being transmitted as part of the record, if links are specified to available images, or if a palm print is

available from the referenced agency. This element may occur multiple times. Only one image is to be coded per <j:PersonPalmPrint> element.

**Palm Print Image (<j:BiometricImage>):**

**GJXDM:** A picture of a biometric sample.

This is a group element reporting the detail of a single image

**Image (<j:BinaryObject.Base64>)**

**GJXDM:** A binary encoding of data, e.g., a binary encoding of a picture, photo, image, graphic, sound, or video.

Optional. If the actual image is contained here, it must be encoded using the Base64 algorithm which transforms binary data into text characters.

**Image Format (<j:BinaryFormatText>)**

**GJXDM:** A format of a binary object, e.g., “image/jpeg”.

The <j:BinaryFormatText> is an optional element containing the format in which the SMT image is stored. Values include “image/gif” “image/tiff” “image/jpeg” “application/postscript” “video/mpeg”. Use “image/wsq” for fingerprint images compressed using the ANSI/NIST/FBI recommended algorithm wavelet scalar quantization. The format is the binary format of the image prior to Base64 encoding, or the format of the image that is the target of a hyperlink (see below).

**Image Size (<j:BinarySizeValue>)**

**GJXDM:** A size of a binary object in kilobytes.

The <j:BinarySizeValue> is an optional element containing the approximate size of the original image (before encoding).

**Palm Print Type (<j:BinaryTypeText>)**

**GJXDM:** A type of binary object that is encoded, e.g., mug shot, driver license picture, audio confession..

The <j:BinaryTypeText> element should contain information about which palm(s) are included in the image. Use a value from the following table:

<i>Palm Print Type</i>	
Right Palm	Left Palm
Right Writer's Palm	Left Writer's Palm
Both Palms	Both Writer's Palms
Unknown	

### **Additional Description or Comment**

(**<j:BinaryDescriptionText**)

**GJXDM:** A textual description of a binary object.

An optional element for reporting additional free-text information or comment about the palm print.

### **Image Hyperlink (<j:BinaryReferenceID)**

**GJXDM:** Identifies a url or file reference of the binary object.

An optional element for reporting a hyperlink to the photo. The hyperlink may be a URL or URI or description of a file location. The actual text value of the hyperlink is reported in an <j:ID> subelement.

Example:

```
<j:BinaryReferenceID>
```

```
<j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
```

```
</j:BinaryReferenceID>
```

### **Date of image (<j:BinaryCaptureDate)**

**GJXDM:** A date on which the data represented by the binary object is captured, e.g., an image taken or an audio recorded..

This element may contain the date on which the palm print image was obtained. The format for the date is CCYY[-MM[-DD]].

### **Image Agency (<j:BinaryCaptureOrganization)**

**GJXDM:** An organization which captured or created a binary.

This is the only required element for an image. Use OrganizationName of "Unknown" if no agency is known. If the agency is reported without image or hyperlink data, then the agency has an image on file and can be contacted to obtain one. Use subelements <j:OrganizationName> and/or <j:OrganizationORIID><ID>.

Examples:

```

<j:PersonPalmPrint>
  <j:BiometricImage>
    <j:BinaryObject.Base64>
      dkriJEORjSo9rj6+/kDIsl...DKRHkdiop
    </j:BinaryObject.Base64>
    <j:BinaryFormatText>image/wsqa</j:BinaryFormatText>
    <j:BinarySizeValue>75</j:BinarySizeValue>
    <j:BinaryTypeText>Right Palm</j:BinaryTypeText>
    <j:BinaryDescriptionText>Grayscale</j:BinaryDescriptionText>
    <j:BinaryReferenceID>
      <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
    </j:BinaryReferenceID>
    <j:BinaryCaptureDate>1999-06-04</j:BinaryCaptureDate>
    <j:BinaryCaptureOrganization>
      <j:OrganizationName>CIB</j:OrganizationName>
      <j:OrganizationORIID>
        <j:ID>WI013415Y</j:ID>
      </j:OrganizationORIID>
    </j:BinaryCaptureOrganization>
  </j:BiometricImage>
</j:PersonPalmPrint>

<j:PersonPalmPrint>
  <j:BinaryTypeText>Both Palms</j:BinaryTypeText>
  <j:BinaryCaptureDate>1999-06-04</j:BinaryCaptureDate>
  <j:BinaryCaptureOrganization>
    <j:OrganizationName>CIB</j:OrganizationName>
    <j:OrganizationORIID>
      <j:ID>WI013415Y</j:ID>
    </j:OrganizationORIID>
  </j:BinaryCaptureOrganization>
</j:PersonPalmPrint>

```

### **Person Medical Description (<j:PersonMedicalDetails>):**

**GJXDM:** A set of details about the medical condition of a person..

This group element contains descriptive information about the record subject's medical condition.

*Note: For all implementations of the rapsheet document, it is suggested that if medical conditions are included, then the document's <rap:Introduction> should include a <rap:Caveat> advising of possible restrictions on the use of medical information.*

### **Condition (<j:PersonMedicalCondition>):**

**GJXDM:** Details about a specific medical condition a person has or experiences..

This element may be repeated to report one or more specific medical conditions. Use the j:@reportedDate attribute to report the date of the associated

condition. The format for the date is CCYY[-MM[-DD]]. Use the <j:MedicalConditionText> subelement to report the specific condition.

Example:

```
<j:PersonMedicalDetails>
  <j:PersonMedicalCondition reportedDate="1964-04-01">
    <j:MedicalConditionText>Suicidal</j:MedicalConditionText>
  </j:PersonMedicalCondition>
  <j:PersonMedicalCondition reportedDate="1969-10-16">
    <j:MedicalConditionText>Diabetic</j:MedicalConditionText>
  </j:PersonMedicalCondition>
</j:PersonMedicalDetails>
```

### **Caution Information (<j:SubjectCautionInformationCaveat>):**

**GJXDM:** Details regarding cautionary messages about an individual.

This element contains a free-text cautionary message concerning the record subject. The <j:SubjectCautionInformationCaveat> element may contain the j:@reportedDate attribute showing the date on which the cautionary message originated. The format for the date is CCYY[-MM[-DD]]. The element may contain the j:@reportingOrganizationText attribute set to the NCIC state two-letter code from which the comment statement was issued; "Unknown" is also an allowable value.

Example:

```
<j:SubjectCautionInformationCaveat
  reportingOrganizationText="CA"
  reportedDate="1995-11-02">Armed and dangerous
</j:SubjectCautionInformationCaveat>
```

### **Offender Notice Information or Comment (<j:SubjectOffenderNoticeCaveat>):**

**GJXDM:** Details about other cautions/caveats/notices related to an offender.

This element contains a free-text description of additional information associated with the record subject, or any other comment. The <j:SubjectOffenderNoticeCaveat> element may contain the j:@reportedDate attribute showing the date on which the notice originated. The format for the date is CCYY[-MM[-DD]]. The element may contain the j:@reportingOrganizationText attribute set to the NCIC state two-letter code from which the notice statement was issued.

Example:

```
<j:SubjectOffenderNoticeCaveat
  reportingOrganizationText="CA"
  reportedDate="1995-11-02">
  Subject of the record is a convicted felon.
</j:SubjectOffenderNoticeCaveat>
```

### 3.3 Cycle (<rap:Cycle>)

Each cycle will be based upon a fingerprint submission that either establishes the subject record or positively identifies the current submission to the existing subject record. Ordinarily, a cycle will begin with an arrest and will contain all the reported information that pertains to that arrest. The incident is the criminal offense that led to the arrest, and for the rapsheet identifies the date of the offense. Cycles may be based upon a fingerprint submission taken by other than arresting agencies, such as prosecutors or custodial agencies. All information contained in the cycle must be related to the fingerprint submission that originally created the cycle. In some instances, multiple fingerprint submissions will exist within a cycle (e.g., arrest and custodial), but they must all relate to the original fingerprint submission.

#### **Earliest Event Date (<rap:CycleEarliestDate>):**

This element contains the earliest date included in the cycle element. The <rap:CycleEarliestDate> must be the earliest of arrest date, arrest disposition, prosecutor disposition, or court disposition dates. Do not use the incident date of offense even though it may be earlier. That is, <rap:CycleEarliestDate> must be the earliest of <j:ActivityDate> under <j:Arrest>, <rap:Prosecution>, <rap:CourtAction>, <rap:Sentencing>, or <rap:Supervision>. The format for the date is CCYY[-MM[-DD]].

Example:

```
<rap:CycleEarliestDate>1995-07-04</rap:CycleEarliestDate>
```

**Cycle Tracking Number (<rap:CycleTrackingID>):** A unique number or alphanumeric identifier assigned to the entire cycle. Usually, this identifier will be the <j:ChargeTrackingID> most representative of the entire cycle. The intent of a tracking number is to associate arrest charges with prosecution charges and with final court charge disposition. A cycle may contain multiple charges, and for many contributing systems a single unique number is related to every charge throughout the cycle. Some systems however, especially court and prosecutor systems, allow case consolidation which results in charges from multiple arrest events (cycles) being disposed together. The intent of this specification is to identify a single number that is representative of the entire cycle, while also permitting each charge to have differing <j:ChargeTrackingID>'s if needed by the reporting system.

Example:

```
<rap:CycleTrackingID>  
  <j:ID>46019527</j:ID>  
</rap:CycleTrackingID>
```

#### **Incident segment <j:Incident>**

**GJXDM:** Details about a criminal or non-criminal activity that occurred.

The GJXDM incident object is included in the rapsheet solely to report the date of criminal offense.

**Date of Offense (<j:ActivityDate>):**

**GJXDM:** A date of an activity that occurs at a singular point in time or a start date of an activity that occurs over a period of time.

This element contains the date of the offense to which the cycle corresponds. The format for the date is CCYY[-MM[-DD]].

Example:

```
<j:ActivityDate>1998-05-28</j:ActivityDate>
```

**Arrest segment (<j:Arrest>):**

**GJXDM:** Details about the apprehension of a subject by a peace official based on an observed or a reported violation of a law or ordinance, an outstanding arrest warrant, or probable cause information.

When the Arrest element is included, a minimum of one <j:ArrestCharge> element is mandatory. All of the law enforcement-initiated charges for this cycle are to be reported in the Arrest segment. If the arresting and booking agencies are different, the Arrest segment contains all of the charges for both agencies.

**Arrest Type (<j:ActivityTypeText>):**

**GJXDM:** A type of activity or process that occurred.

This element contains the guidelines under which it is anticipated the subject will be processed, such as, Adult, Juvenile, Juvenile as Adult, etc.

Example:

```
<j:ActivityTypeText>Adult</j:ActivityTypeText>
```

**Arrest Date (<j:ActivityDate>):**

**GJXDM:** A date of an activity that occurs at a singular point in time or a start date of an activity that occurs over a period of time.

This element contains the date of arrest. The format for the date is CCYY[-MM[-DD]]. (EFTS 2.045)

Example:

```
<j:ActivityDate>1998-05-30</j:ActivityDate>
```

**Arrest Comments (<j:ActivityCommentText>):**

**GJXDM:** A note or comment about an activity.

This element contains information supplemental or ancillary to the other data specified in the Arrest segment.

Example:

```
<j:ActivityCommentText>Resisted  
arrest</j:ActivityCommentText>
```

**Arrest Case Number (<j:ArrestAgencyRecordID>):**

**GJXDM:** A records management system number of the originating case agency for an arrest. This may be the booking number or the Records Management System number. Sometimes referred to as Originating Case Agency (OCA) number.

This element contains the case number assigned by the arresting agency. This element, called the Originating Case Agency number (OCA) in the III, can be used to enter an AFIS or Process Control Number from the arrest fingerprint card.

*Note: see Booking segment for reporting Booking agency OCA.*

Example:

```
<j:ArrestAgencyRecordID>  
  <j:ID>1998AF002354</j:ID>  
</j:ArrestAgencyRecordID>
```

### **Subject of Arrest (<j:ArrestSubject>):**

**GJXDM:** A person who is arrested.

This group element contains the name and local identifier for the arrested person.

#### **Subject Name (<j:PersonName>):**

**GJXDM:** A name by which a person is known.

This element should include all names for the record subject, including “also known as” (AKA) alias names. Use the <j:PersonFullName> subelement.

Example:

```
<j:PersonName>  
  <j:PersonFullName>Mitch Doherty</j:PersonFullName>  
</j:PersonName>  
<j:PersonName>  
  <j:PersonFullName>Michael Doherty</j:PersonFullName>  
</j:PersonName>
```

#### **Arrest Offender Identification Number (<j:SubjectID>):**

**GJXDM:** An assigned number or string that identifies a subject..

This element contains a unique identification number assigned to an arrest subject by the local arresting agency, much like the SID number assigned at the state level or the FNU assigned at the federal level.

Example:

```
<j:SubjectID>  
  <j:ID>48204395</j:ID>  
</j:SubjectID>
```

### **Arrest Charge (<j:ArrestCharge>):**

**GJXDM:** A formal allegation of a violation of a statute and/or ordinance in association with an arrest.



At least one Arrest Charge element must be included in the Arrest segment.

**Charge Number (<j:ChargeID>):**

**GJXDM:** A unique identifying number assigned to a particular charge by an arresting agency, prosecuting attorney, or a court for case management purposes.

This element contains a unique identifying number assigned to the particular charge for case management purposes.

Example:

```
<j:ChargeID>
  <j:ID>94D002356</j:ID>
</j:ChargeID>
```

**Charge Literal (<j:ChargeText>):**

**GJXDM:** A plain language description of a charge.

This element contains text describing the charge.

Example:

```
<j:ChargeText>Robbery
</j:ChargeText>
```

**Charge Sequence Number (<j:ChargeSequenceID>):**

**GJXDM:** A sequentially assigned number for charge tracking purposes (for example, the first charge under the arrest might be assigned Charge Sequence ID 1, the second 2, and so forth).

This element contains a sequentially assigned number for charge tracking purposes (for example, the first charge under the arrest might be assigned Charge Sequence Number 1, the second 2, and so forth).

Example:

```
<j:ChargeSequenceID>
  <j:ID>01</j:ID>
</j:ChargeSequenceID>
```

**Charge Tracking Number (<j:ChargeTrackingID>):**

**GJXDM:** A unique identifying number assigned to an entire set of charges for an arrest. Different numbers may appear in the set if cases have been consolidated.

This element contains a unique identifying number assigned to the entire set of charges for this arrest/cycle. Different numbers may appear in the set if cases have been consolidated.

Example:

```
<j:ChargeTrackingID>
  <j:ID>AD486</j:ID>
</j:ChargeTrackingID>
```

**Charge Comments (<j:ChargeDescriptionText>):**

**GJXDM:** A plain language description of the charge.

Use this element to report information supplemental or ancillary to the other data specified in the Arrest Charge element.

Example:

```
<j:ChargeDescriptionText>Referred for clinical  
evaluation</j:ChargeDescriptionText>
```

**Charge Classification (<j:ChargeClassification>):**

**GJXDM:** Details that further describes and classifies a charge..

This group element contains supplemental information about the charge.

**Inchoate Charge (<j:ChargeApplicabilityText>):**

**GJXDM:** A degree of involvement a person is being charged with committing in an offense. Sometimes referred to as A/C/S, InchoateCharge. Example: Solicitation; Conspiracy; Attempt; Accessory Before the Fact..

This optional element may be used to report charges that describe the subject's involvement in the offense. The following are valid values to be entered in this element: Solicitation; Conspiracy; Attempt; Accomplice.

Example:

```
<j:ChargeApplicabilityText>Attempt  
</j:ChargeApplicabilityText>
```

**Charge Reducing Factor (<j:ChargeReducingFactorText>):**

**GJXDM:** A factor which may make a charge less serious or limit the penalty (e.g., violations observed by photo-radar stations in some states).

This element may be used to report factors that make the charge less serious.

Example:

```
<j:ChargeReducingFactorText>First offense  
</j:ChargeReducingFactorText>
```

**Charge Enhancing Factor (<j:ChargeSpecialAllegationText>):**

**GJXDM:** A factor that has enhanced a charge, making it a more serious offense.

This element may be used to report factors that make the charge more serious.

Example:

```
<j:ChargeSpecialAllegationText>Elderly  
victim</j:ChargeSpecialAllegationText>
```

**Charge Severity (<j:ChargeSeverityText>):**

**GJXDM:** A level of severity of a charge, e.g., Felony; Misdemeanor; Petty Misdemeanor; Other; Unknown.

This element contains the charge severity. The following are valid values to be entered in this element: Felony; Misdemeanor; Local Ordinance; Other; Unknown.

Example:

```
<j:ChargeSeverityText>Felony</j:ChargeSeverityText>
```

**Counts (<j:ChargeCountQuantity>):**

**GJXDM:** A number of times a person is charged with committing the same crime.

This is an optional, numeric element minimum length one character, maximum length three characters. This element can be used to show that the subject was charged multiple times for the same crime.

Example:

```
<j:ChargeCountQuantity>4</j:ChargeCountQuantity>
```

**Arrest Action Literal (<j:ChargeDisposition>):**

**GJXDM:** Details about the results or processing of a charge.

This element is optional, and it is expected that it would not be commonly used in the Arrest segment. If omitted, it is assumed that the arrest details were referred for prosecution. If the arrest charges are dismissed by the law enforcement agency, or dropped, or the case is not referred for prosecution, then this element should be used to report a final disposition for the cycle.

This element must contain a value for the @commentText attribute most closely associated with the disposition from the following table:

<i>Disposition Type</i>	
Acquitted	Deferred
Convicted	Failure To Appear
Revocation	Dismissed
Charges Dropped	Not Prosecuted
Other	Unknown

**Disposition Date (<j:ChargeDispositionDate>):**

**GJXDM:** A date of a disposition of a charge.

This element reports the date on which the action was taken. The format for the date is CCYY[-MM[-DD]].

## Disposition

(**<j:ChargeDispositionDescriptionText>**):

**GJXDM:** A description of the result of processing a charge. For an arrest charge, this may be a law enforcement agency action, such as custody only, turned over to another agency, referred or not referred for prosecution. For a court case charge, this may be dismissal, or any penalties, sanctions, or sentence assessed as a result of a finding or negotiated resolution in the case.

This element contains a free-text description of law enforcement agency action, such as custody only, turned over to another agency, not referred for prosecution, or referred for prosecution.

Example:

```
<j:ChargeDisposition commentText="Charges Dropped">
  <j:ChargeDispositionDate>1996-07-07
  </j:ChargeDispositionDate>
  <j:ChargeDispositionDescriptionText>
    Case closed without referral for prosecution.
  </j:ChargeDispositionDescriptionText>
</j:ChargeDisposition>
```

**Charge NCIC Code (<j:ChargeReporting>):**

**GJXDM:** Information that is sent to local, state, or federal repositories describing the nature of a charge in terms of official codes, terms, and fields.

**Charge NCIC Code (<j:ChargeNCICCode>):**

**GJXDM:** A code that identifies an offense within the National Crime Information Center (NCIC) system.

This element contains the NCIC offense code. NCIC offense codes are listed in the NCIC 2000 Code Manual.

Example:

```
<j:ChargeReporting>
  <j:ChargeNCICCode>8312</j:ChargeNCICCode>
</j:ChargeReporting>
```

**State or Local Offense Information (<j:ChargeStatute>):**

**GJXDM:** A unique identifier of a law, rule, or ordinance within a jurisdiction that a person is accused of violating.

**Statute Number (<j:StatuteCodeID>):**

**GJXDM:** An identifier of a set of laws for a particular jurisdiction. Sometimes referred to as a code book, legal code, native state code, or law.

This element contains the statute number.

*Note: Implementers are advised to avoid use of the "\$" subsection character.*

**State Offense Code (<j:StatuteOffenseID>):**

**GJXDM:** A number or code that identifies a criminal offense within a code book. Sometimes referred to as offense code, ordinance number.

This element contains the offense code from the state.

**Charge State (<j:StatuteJurisdiction>):**

**GJXDM:** Details about an area in which a statute applies, e.g., city, state, country.

Use the <j:LocationStateName> subelement to report the 2-character state abbreviation.

**Example:**

```
<j:ChargeStatute>
  <j:StatuteCodeID>
    <j:ID>943.2(e)10</j:ID>
  </j:StatuteCodeID>
  <j:StatuteOffenseID>
    <j:ID>20836</j:ID>
  </j:StatuteOffenseID>
  <j:StatuteJurisdiction>
    <j:LocationStateName>WI
  </j:LocationStateName>
  </j:StatuteJurisdiction>
</j:ChargeStatute>
```

**Arrest Agency (<j:ArrestAgency>):**

**GJXDM:** An agency which employs the arresting official.

This element contains information about the arresting agency responsible for one or more of the charges in the Arrest segment. One or both of the subelements <j:OrganizationName> or <j:OrganizationORIID> must be present.

**Example:**

```
<j:ArrestAgency>
  <j:OrganizationName>CIB</j:OrganizationName>
  <j:OrganizationORIID>
    <j:ID>WI013415Y</j:ID>
  </j:OrganizationORIID>
</j:ArrestAgency>
```

**Booking segment (<j:Booking>):**

**GJXDM:** Details about an administrative step taken after an arrested subject is brought to a police station or detention facility, which involves entry of the subject's name and other relevant facts on the police "blotter", and which may also include photographing, fingerprinting, and the like.

This optional segment, and group element, contains information about the booking agency, if different from the arresting agency. This segment may be omitted if the booking and arrest agencies are identical. Often, a booking agency is responsible for obtaining fingerprint impressions and contributing records to a state identification bureau.

Booking agencies may be responsible for some or all of the charges in the <j:Arrest> segment.

**Booking Case Number (<j:BookingAgencyRecordID>):**

**GJXDM** A booking identifier of the originating case agency. This identifier may be the booking number or the Originating Agency Report (Incident) number or a combination of both. Also known as the Originating Case Agency (OCA) number.

This element contains the case number assigned by the booking agency. This element, called the Originating Case Agency number (OCA) in the III, can be used to enter an AFIS or Process Control Number from the arrest fingerprint card. *Note: see Arrest segment for reporting the Arresting agency's OCA.*

Example:

```
<j:BookingAgencyRecordID>  
  <j:ID>1998AF002355</j:ID>  
</j:BookingAgencyRecordID>
```

**Booking Agency (<j:BookingAgency>):**

**GJXDM:** An organization which processes a booking.

This element contains information about the booking agency. . One or both of the subelements <j:OrganizationName> or <j:OrganizationORIID> must be present.

Example:

```
<j:BookingAgency>  
  <j:OrganizationName>CIB</j:OrganizationName>  
  <j:OrganizationORIID>  
    <j:ID>WI013415Y</j:ID>  
  </j:OrganizationORIID>  
</j:BookingAgency>
```

**Prosecution Segment (<rap:Prosecution>):**

When the Prosecution segment element is included, a minimum of one Prosecutor Charge element is mandatory. If, for this cycle, any of the charges are filed with a court, then the Prosecution segment is optional. This segment should be used when the Prosecutor represents the final stage in this cycle – no prosecution of any charge occurs. This segment may be used to report an intermediate set of charges filed with a court by the Prosecutor.

**Prosecution Filing or Closing Date (<j:ActivityDate>):**

**GJXDM:** A date of an activity that occurs at a singular point in time or a start date of an activity that occurs over a period of time.

This element contains the date the prosecutor first files charges with a court, or the date the prosecutor closes the case by declining to prosecute all charges referred by law enforcement. The format for the date is CCYY[-MM[-DD]]. (EFTS 2.045)

Example:

```
<j:ActivityDate>1998-05-30</j:ActivityDate>
```

**Prosecution Comments (<j:ActivityCommentText>):****GJXDM:** A note or comment about an activity.

This element contains information supplemental or ancillary to the other data specified in the Prosecution segment.

Example:

```
<j:ActivityCommentText>Repeat
offender.</j:ActivityCommentText>
```

**Prosecutor Case Number (<rap:ProsecutionAgencyRecordID>):**

This element contains the case number assigned by the prosecuting agency. This element is also called the Originating Case Agency number (OCA).

Example:

```
<rap:ProsecutionAgencyRecordID>
  <j:ID>1998AF002356</j:ID>
</rap:ProsecutionAgencyRecordID>
```

**Subject of Prosecution (<rap:ProsecutionSubject>):**

This group element contains the name and local identifier for the prosecuted person.

**Subject Name (<j:PersonName>):****GJXDM:** A name by which a person is known.

This element should include all names for the record subject, including “also known as” (AKA) alias names. Use the <j:PersonFullName> subelement.

**Prosecution Person Identification Number (<j:SubjectID>):****GJXDM:** An assigned number or string that identifies a subject..

This element contains a unique identification number assigned to a subject by the prosecutor, much like the SID number assigned at the state level or the FNU assigned at the federal level.

Example:

```
<rap:ProsecutionSubject>
  <j:PersonName>
    <j:PersonFullName>Mitch Doherty</j:PersonFullName>
  </j:PersonName>
  <j:PersonName>
    <j:PersonFullName>Michael Doherty</j:PersonFullName>
  </j:PersonName>
  <j:SubjectID>
    <j:ID>48204396</j:ID>
  </j:SubjectID>
</rap:ProsecutionSubject>
```

**Prosecution Charge (<j:ProsecutionCharge>):**

**GJXDM:** A charge filed by a prosecuting attorney. May be different from or the same as the original arrest charge.

At least one Prosecution Charge element must be included in the Prosecution segment.

**Charge Number (<j:ChargeID>):**

**GJXDM:** A unique identifying number assigned to a particular charge by an arresting agency, prosecuting attorney, or a court for case management purposes.

This element contains a unique identifying number assigned to the particular charge for case management purposes.

Example:

```
<j:ChargeID>  
  <j:ID>94D002356</j:ID>  
</j:ChargeID>
```

**Charge Literal (<j:ChargeText>):**

**GJXDM:** A plain language description of a charge.

This element contains text describing the charge.

Example:

```
<j:ChargeText>Burglary  
</j:ChargeText>
```

**Charge Sequence Number (<j:ChargeSequenceID>):**

**GJXDM:** A sequentially assigned number for charge tracking purposes (for example, the first charge under the arrest might be assigned Charge Sequence ID 1, the second 2, and so forth).

This element contains a sequentially assigned number for charge tracking purposes (for example, the first charge under the arrest might be assigned Charge Sequence Number 1, the second 2, and so forth).

Example:

```
<j:ChargeSequenceID>  
  <j:ID>01</j:ID>  
</j:ChargeSequenceID>
```

**Charge Tracking Number (<j:ChargeTrackingID>):**

**GJXDM:** A unique identifying number assigned to an entire set of charges for an arrest. Different numbers may appear in the set if cases have been consolidated.

This element contains a unique identifying number assigned to the entire set of charges for this cycle. Different numbers may appear in the set if cases have been consolidated.

Example:

```
<j:ChargeTrackingID>  
  <j:ID>AD486</j:ID>  
</j:ChargeTrackingID>
```



**Charge Comments (<j:ChargeDescriptionText>):**

**GJXDM:** A plain language description of the charge.

Use this element to report information supplemental or ancillary to the other data specified in the Prosecutor Charge element.

Example:

```
<j:ChargeDescriptionText>Referred for clinical  
evaluation</j:ChargeDescriptionText>
```

**Charge Classification (<j:ChargeClassification>):**

**GJXDM:** Details that further describes and classifies a charge..

This group element contains supplemental information about the charge.

**Inchoate Charge (<j:ChargeApplicabilityText>):**

**GJXDM:** A degree of involvement a person is being charged with committing in an offense. Sometimes referred to as A/C/S, InchoateCharge. Example: Solicitation; Conspiracy; Attempt; Accessory Before the Fact..

This optional element may be used to report charges that describe the subject's involvement in the offense. The following are valid values to be entered in this element: Solicitation; Conspiracy; Attempt; Accomplice.

Example:

```
<j:ChargeApplicabilityText>Attempt  
</j:ChargeApplicabilityText>
```

**Charge Reducing Factor (<j:ChargeReducingFactorText>):**

**GJXDM:** A factor which may make a charge less serious or limit the penalty (e.g., violations observed by photo-radar stations in some states).

This element may be used to report factors that make the charge less serious.

Example:

```
<j:ChargeReducingFactorText>First offense  
</j:ChargeReducingFactorText>
```

**Charge Enhancing Factor (<j:ChargeSpecialAllegationText>):**

**GJXDM:** A factor that has enhanced a charge, making it a more serious offense.

This element may be used to report factors that make the charge more serious.

Example:

```
<j:ChargeSpecialAllegationText>Elderly  
victim</j:ChargeSpecialAllegationText>
```

**Charge Severity (<j:ChargeSeverityText>):**

**GJXDM:** A level of severity of a charge, e.g., Felony; Misdemeanor; Petty Misdemeanor; Other; Unknown.

This element contains the charge severity. The following are valid values to be entered in this element: Felony; Misdemeanor; Local Ordinance; Other; Unknown.

Example:

```
<j:ChargeSeverityText>Felony</j:ChargeSeverityText>
```

**Counts (<j:ChargeCountQuantity>):**

**GJXDM:** A number of times a person is charged with committing the same crime.

This is an optional, numeric element minimum length one character, maximum length three characters. This element can be used to show that the subject was charged multiple times for the same crime.

Example:

```
<j:ChargeCountQuantity>4</j:ChargeCountQuantity>
```

**Prosecution Action Literal (<j:ChargeDisposition>):**

**GJXDM:** Details about the results or processing of a charge.

This element is optional, and it is expected that it would usually be used to report charges dismissed, dropped, or delined for prosecution. If the all the charges referred by the arresting agency are dismissed by the prosecutor, and no additional charges are added, then this element should be used to report a final disposition for the cycle.

This element must contain a value for the @commentText attribute most closely associated with the disposition from the following table:

<i>Disposition Type</i>	
Acquitted	Deferred
Convicted	Failure To Appear
Revocation	Dismissed
Charges Dropped	Not Prosecuted
Other	Unknown

**Disposition Date (<j:ChargeDispositionDate>):**

**GJXDM:** A date of a disposition of a charge.

This element reports the date on which the action was taken. The format for the date is CCYY[-MM[-DD]].

### **Disposition**

(**<j:ChargeDispositionDescriptionText>**):

**GJXDM:** A description of the result of processing a charge. For an arrest charge, this may be a law enforcement agency action, such as custody only, turned over to another agency, referred or not referred for prosecution. For a court case charge, this may be dismissal, or any penalties, sanctions, or sentence assessed as a result of a finding or negotiated resolution in the case.

This element contains a free-text description of prosecutor action such as not referred for prosecution, or filed for court action.

Example:

```
<j:ChargeDisposition commentText="Not Prosecuted">
  <j:ChargeDispositionDate>1996-07-07
  </j:ChargeDispositionDate>
  <j:ChargeDispositionDescriptionText>
    Case closed without court filing.
  </j:ChargeDispositionDescriptionText>
</j:ChargeDisposition>
```

**Charge NCIC Code (<j:ChargeReporting>):**

**GJXDM:** Information that is sent to local, state, or federal repositories describing the nature of a charge in terms of official codes, terms, and fields.

**Charge NCIC Code (<j:ChargeNCICCode>):**

**GJXDM:** A code that identifies an offense within the National Crime Information Center (NCIC) system.

This element contains the NCIC offense code. NCIC offense codes are listed in the NCIC 2000 Code Manual.

Example:

```
<j:ChargeReporting>
  <j:ChargeNCICCode>8312</j:ChargeNCICCode>
</j:ChargeReporting>
```

**State or Local Offense Information (<j:ChargeStatute>):**

**GJXDM:** A unique identifier of a law, rule, or ordinance within a jurisdiction that a person is accused of violating.

**Statute Number (<j:StatuteCodeID>):**

**GJXDM:** An identifier of a set of laws for a particular jurisdiction. Sometimes referred to as a code book, legal code, native state code, or law.

This element contains the statute number.

*Note: Implementers are advised to avoid use of the "\$" subsection character.*

**State Offense Code (<j:StatuteOffenseID>):**

**GJXDM:** A number or code that identifies a criminal offense within a code book. Sometimes referred to as offense code, ordinance number.

This element contains the offense code from the state.

**Charge State (<j:StatuteJurisdiction>):**

**GJXDM:** Details about an area in which a statute applies, e.g., city, state, country.

Use the <j:LocationStateName> subelement to report the 2-character state abbreviation.

**Example:**

```
<j:ChargeStatute>
  <j:StatuteCodeID>
    <j:ID>943.2(e)10</j:ID>
  </j:StatuteCodeID>
  <j:StatuteOffenseID>
    <j:ID>20836</j:ID>
  </j:StatuteOffenseID>
  <j:StatuteJurisdiction>
    <j:LocationStateName>WI
  </j:LocationStateName>
  </j:StatuteJurisdiction>
</j:ChargeStatute>
```

**Prosecution Agency (<rap:ProsecutionAgency>):**

This element contains information about the prosecuting agency responsible for one or more of the charges in the Prosecution segment. One or both of the subelements <j:OrganizationName> or <j:OrganizationORIID> must be present.

**Example:**

```
<rap:ProsecutionAgency>
  <j:OrganizationName>CIB</j:OrganizationName>
  <j:OrganizationORIID>
    <j:ID>WI013415Y</j:ID>
  </j:OrganizationORIID>
</rap:ProsecutionAgency>
```

**Court Segment (<rap:CourtAction>):**

When the Court segment element is included, a minimum of one Court Charge element is mandatory. This segment is intended primarily to report a final disposition on charges. Ideally, the <j:ChargeSequenceID> has been used to tie Arrest, Prosecution, and Court charges together (although it is recognized that many systems cannot make that link). Thus, ArrestCharge (01) of Robbery may show up in the Prosecutor segment as ProsecutorCharge (01) Burglary, and in the Court segment as CourtCharge (01) Trespassing conviction.

**Court Final Disposition Date (<j:ActivityDate>):**

**GJXDM:** A date of an activity that occurs at a singular point in time or a start date of an activity that occurs over a period of time.

This element contains the date of the court findings, the date on which all case charges have been adjudicated. The format for the date is CCYY[-MM[-DD]]. (EFTS 2.045)

Example:

```
<j:ActivityDate>1998-05-30</j:ActivityDate>
```

**Court Comments (<j:ActivityCommentText>):**

**GJXDM:** A note or comment about an activity.

This element contains information supplemental or ancillary to the other data specified in the Court segment.

Example:

```
<j:ActivityCommentText>Jury trial
waived.</j:ActivityCommentText>
```

**Court Case Number (<rap:CourtRecordID>):**

This element contains the case number assigned by the court. This element is also called the Originating Case Agency number (OCA).

Example:

```
<rap:CourtRecordID>
  <j:ID>1998AF002356</j:ID>
</rap:CourtRecordID>
```

**Defendant (<rap:CourtSubject>):**

This group element contains the name and local identifier for the court defendant.

**Subject Name (<j:PersonName>):**

**GJXDM:** A name by which a person is known.

This element should include all names for the record subject, including “also known as” (AKA) alias names. Use the <j:PersonFullName> subelement.

**Court Person Identification Number (<j:SubjectID>):**

**GJXDM:** An assigned number or string that identifies a subject..

This element contains a unique identification number assigned to a defendant by the court, much like the SID number assigned at the state level or the FNU assigned at the federal level.

Example:

```
<rap:CourtSubject>
  <j:PersonName>
    <j:PersonFullName>Mitch Doherty</j:PersonFullName>
  </j:PersonName>
  <j:PersonName>
```

```
<j:PersonFullName>Michael Doherty</j:PersonFullName>
</j:PersonName>
<j:SubjectID>
  <j:ID>48204397</j:ID>
</j:SubjectID>
</rap:CourtSubject>
```

### **Court Charge (<j:CourtCharge>):**

**GJXDM:** A charge a person is tried for in court. May be different from or the same as its predecessor arrest and prosecution charges.

At least one Court Charge element must be included in the Court segment.

#### **Charge Number (<j:ChargeID>):**

**GJXDM:** A unique identifying number assigned to a particular charge by an arresting agency, prosecuting attorney, or a court for case management purposes.

This element contains a unique identifying number assigned to the particular charge for case management purposes.

Example:

```
<j:ChargeID>
  <j:ID>94D002356</j:ID>
</j:ChargeID>
```

#### **Charge Literal (<j:ChargeText>):**

**GJXDM:** A plain language description of a charge.

This element contains text describing the charge.

Example:

```
<j:ChargeText>Trespassing
</j:ChargeText>
```

#### **Charge Sequence Number (<j:ChargeSequenceID>):**

**GJXDM:** A sequentially assigned number for charge tracking purposes (for example, the first charge under the arrest might be assigned Charge Sequence ID 1, the second 2, and so forth).

This element contains a sequentially assigned number for charge tracking purposes (for example, the first charge under the arrest might be assigned Charge Sequence Number 1, the second 2, and so forth).

Example:

```
<j:ChargeSequenceID>
  <j:ID>01</j:ID>
</j:ChargeSequenceID>
```

#### **Charge Tracking Number (<j:ChargeTrackingID>):**

**GJXDM:** A unique identifying number assigned to an entire set of charges for an arrest. Different numbers may appear in the set if cases have been consolidated.

This element contains a unique identifying number assigned to the entire set of charges for this cycle. Different numbers may appear in the set if cases have been consolidated.

Example:

```
<j:ChargeTrackingID>  
  <j:ID>AD486</j:ID>  
</j:ChargeTrackingID>
```

### **Charge Comments (<j:ChargeDescriptionText>):**

**GJXDM:** A plain language description of the charge.

Use this element to report information supplemental or ancillary to the other data specified in the Court Charge element.

Example:

```
<j:ChargeDescriptionText>Referred for clinical  
evaluation</j:ChargeDescriptionText>
```

### **Charge Classification (<j:ChargeClassification>):**

**GJXDM:** Details that further describes and classifies a charge..

This group element contains supplemental information about the charge.

#### **Inchoate Charge (<j:ChargeApplicabilityText>):**

**GJXDM:** A degree of involvement a person is being charged with committing in an offense. Sometimes referred to as A/C/S, InchoateCharge. Example: Solicitation; Conspiracy; Attempt; Accessory Before the Fact..

This optional element may be used to report charges that describe the subject's involvement in the offense. The following are valid values to be entered in this element: Solicitation; Conspiracy; Attempt; Accomplice.

Example:

```
<j:ChargeApplicabilityText>Attempt  
</j:ChargeApplicabilityText>
```

#### **Charge Reducing Factor (<j:ChargeReducingFactorText>):**

**GJXDM:** A factor which may make a charge less serious or limit the penalty (e.g., violations observed by photo-radar stations in some states).

This element may be used to report factors that make the charge less serious.

Example:

```
<j:ChargeReducingFactorText>First offense  
</j:ChargeReducingFactorText>
```

#### **Charge Enhancing Factor (<j:ChargeSpecialAllegationText>):**

**GJXDM:** A factor that has enhanced a charge, making it a more serious offense.

This element may be used to report factors that make the charge more serious.

Example:

```
<j:ChargeSpecialAllegationText>Elderly  
victim</j:ChargeSpecialAllegationText>
```

### **Charge Severity (<j:ChargeSeverityText>):**

**GJXDM:** A level of severity of a charge, e.g., Felony; Misdemeanor; Petty Misdemeanor; Other; Unknown.

This element contains the charge severity. The following are valid values to be entered in this element: Felony; Misdemeanor; Local Ordinance; Other; Unknown.

Example:

```
<j:ChargeSeverityText>Felony</j:ChargeSeverityText>
```

### **Counts (<j:ChargeCountQuantity>):**

**GJXDM:** A number of times a person is charged with committing the same crime.

This is an optional, numeric element minimum length one character, maximum length three characters. This element can be used to show that the subject was charged multiple times for the same crime.

Example:

```
<j:ChargeCountQuantity>4</j:ChargeCountQuantity>
```

### **Court Action Literal (<j:ChargeDisposition>):**

**GJXDM:** Details about the results or processing of a charge.

This element is optional, but expected. The primary purpose for reporting charges under the Court segment is to report a charge disposition.

This element must contain a value for the @commentText attribute most closely associated with the disposition from the following table:

<i>Disposition Type</i>	
Acquitted	Deferred
Convicted	Failure To Appear
Revocation	Dismissed
Charges Dropped	Not Prosecuted
Other	Unknown

### **Disposition Date (<j:ChargeDispositionDate>):**

**GJXDM:** A date of a disposition of a charge.



This element reports the date on which the action was taken. The format for the date is CCYY[-MM[-DD]].

### **Disposition**

(**<j:ChargeDispositionDescriptionText>**):

**GJXDM:** A description of the result of processing a charge. For an arrest charge, this may be a law enforcement agency action, such as custody only, turned over to another agency, referred or not referred for prosecution. For a court case charge, this may be dismissal, or any penalties, sanctions, or sentence assessed as a result of a finding or negotiated resolution in the case.

This element contains a free-text description of a court finding, action, or verdict, such as guilty or dismissed.

Example:

```
<j:ChargeDisposition commentText="Convicted">
  <j:ChargeDispositionDate>1996-07-07
  </j:ChargeDispositionDate>
  <j:ChargeDispositionDescriptionText>
    Guilty.
  </j:ChargeDispositionDescriptionText>
</j:ChargeDisposition>
```

**Charge NCIC Code (<j:ChargeReporting>):**

**GJXDM:** Information that is sent to local, state, or federal repositories describing the nature of a charge in terms of official codes, terms, and fields.

**Charge NCIC Code (<j:ChargeNCICCode>):**

**GJXDM:** A code that identifies an offense within the National Crime Information Center (NCIC) system.

This element contains the NCIC offense code. NCIC offense codes are listed in the NCIC 2000 Code Manual.

Example:

```
<j:ChargeReporting>
  <j:ChargeNCICCode>8312</j:ChargeNCICCode>
</j:ChargeReporting>
```

**State or Local Offense Information (<j:ChargeStatute>):**

**GJXDM:** A unique identifier of a law, rule, or ordinance within a jurisdiction that a person is accused of violating.

**Statute Number (<j:StatuteCodeID>):**

**GJXDM:** An identifier of a set of laws for a particular jurisdiction. Sometimes referred to as a code book, legal code, native state code, or law.

This element contains the statute number.

*Note: Implementers are advised to avoid use of the "\$" subsection character.*

**State Offense Code (<j:StatuteOffenseID>):**

**GJXDM:** A number or code that identifies a criminal offense within a code book. Sometimes referred to as offense code, ordinance number.

This element contains the offense code from the state.

**Charge State (<j:StatuteJurisdiction>):**

**GJXDM:** Details about an area in which a statute applies, e.g., city, state, country.

Use the <j:LocationStateName> subelement to report the 2-character state abbreviation.

**Example:**

```
<j:ChargeStatute>
  <j:StatuteCodeID>
    <j:ID>943.2(e)10</j:ID>
  </j:StatuteCodeID>
  <j:StatuteOffenseID>
    <j:ID>20836</j:ID>
  </j:StatuteOffenseID>
  <j:StatuteJurisdiction>
    <j:LocationStateName>WI
  </j:LocationStateName>
  </j:StatuteJurisdiction>
</j:ChargeStatute>
```

**Court (<j:Court>):**

**GJXDM:** Details about a court or a unit of a court responsible for trying justice proceedings.

This element contains information about the court responsible for the final disposition on charges in the Court segment. One or both of the subelements <j:OrganizationName> or <j:OrganizationORIID> must be present.

**Example:**

```
<j:Court>
  <j:OrganizationName>Portage County Circuit Court
  </j:OrganizationName>
  <j:OrganizationORIID>
    <j:ID>WI014400Y</j:ID>
  </j:OrganizationORIID>
</j:Court>
```

**Sentencing segment (<rap:Sentencing>):**

This element contains the details of the sentence imposed by a court for conviction on charges in this cycle.

**Sentencing Date (<j:ActivityDate>):**

**GJXDM:** A date of an activity that occurs at a singular point in time or a start date of an activity that occurs over a period of time.

This element contains the date of the court sentence. The format for the date is CCYY[-MM[-DD]]. (EFTS 2.045)

Example:

```
<j:ActivityDate>1998-05-30</j:ActivityDate>
```

**Sentence Comments (<j:ActivityCommentText>):**

**GJXDM:** A note or comment about an activity.

This element contains information supplemental or ancillary to the other data specified in the Sentencing segment. Comments may include a description of the sentence being imposed, e.g., probation, incarceration, public service.

Example:

```
<j:ActivityCommentText>Psychiatric evaluation ordered
</j:ActivityCommentText>
```

**Court Case Number (<rap:CourtRecordID>):**

This element contains the case number assigned by the court. This element is also called the Originating Case Agency number (OCA).

Example:

```
<rap:CourtRecordID>
  <j:ID>1998AF002356</j:ID>
</rap:CourtRecordID>
```

**Sentence (<j:Sentence>):**

**GJXDM:** Details about a punishment resulting from conviction of charges in a court case.

**Convicted Offense (<j:SentenceCharge>):**

**GJXDM:** A specific charge in a court case resulting in a sentence.

The details of the charge(s) may optionally be reported here.

**Charge Number (<j:ChargeID>):**

**GJXDM:** A unique identifying number assigned to a particular charge by an arresting agency, prosecuting attorney, or a court for case management purposes.

This element contains a unique identifying number assigned to the particular charge for case management purposes.

Example:

```
<j:ChargeID>
  <j:ID>94D002356</j:ID>
</j:ChargeID>
```

**Charge Literal (<j:ChargeText>):**

**GJXDM:** A plain language description of a charge.

This element contains text describing the charge.

Example:

```
<j:ChargeText>Trespassing
</j:ChargeText>
```

**Charge Sequence Number (<j:ChargeSequenceID>):**

**GJXDM:** A sequentially assigned number for charge tracking purposes (for example, the first charge under the arrest might be assigned Charge Sequence ID 1, the second 2, and so forth).

This element contains a sequentially assigned number for charge tracking purposes (for example, the first charge under the arrest might be assigned Charge Sequence Number 1, the second 2, and so forth).

Example:

```
<j:ChargeSequenceID>  
  <j:ID>01</j:ID>  
</j:ChargeSequenceID>
```

**Charge Tracking Number (<j:ChargeTrackingID>):**

**GJXDM:** A unique identifying number assigned to an entire set of charges for an arrest. Different numbers may appear in the set if cases have been consolidated.

This element contains a unique identifying number assigned to the entire set of charges for this cycle. Different numbers may appear in the set if cases have been consolidated.

Example:

```
<j:ChargeTrackingID>  
  <j:ID>AD486</j:ID>  
</j:ChargeTrackingID>
```

**Sentence Detail (<j:SentenceDescriptionText>):**

**GJXDM:** A description of the sentence being imposed, e.g., probation, incarceration, public service.

This element contains a literal explanation of the court sentence including time and dollar amounts. Multiple <j:SentenceDescriptionText> elements may be used.

Example:

```
<j:SentenceDescriptionText>3Y Prison  
</j:SentenceDescriptionText>  
<j:SentenceDescriptionText>$500 fine and costs  
</j:SentenceDescriptionText>
```

**Court (<j:Court>):**

**GJXDM:** Details about a court or a unit of a court responsible for trying justice proceedings.

This element contains information about the court responsible for the sentencing. One or both of the subelements <j:OrganizationName> or <j:OrganizationORIID> must be present.

Example:

```
<j:Court>  
  <j:OrganizationName>Portage County Circuit Court  
  </j:OrganizationName>
```

```
<j:OrganizationORIID>
  <j:ID>WI014400Y</j:ID>
</j:OrganizationORIID>
</j:Court>
```

### **Supervision Segment (<rap:Supervision>):**

This element contains the details of custody events involving the rapsheet subject. Sometimes a Supervision segment will be included in a cycle that also contains Arrest and Court segments. It is possible, however, for a Supervision segment to be the only segment in a cycle.

#### **Supervision Date (<j:ActivityDate>):**

**GJXDM:** A date of an activity that occurs at a singular point in time or a start date of an activity that occurs over a period of time.

This element contains the date the subject was admitted for supervision or custody. The format for the date is CCYY[-MM[-DD]]. (EFTS 2.045)

Example:

```
<j:ActivityDate>1998-05-30</j:ActivityDate>
```

#### **Supervision Comments (<j:ActivityCommentText>):**

**GJXDM:** A note or comment about an activity.

This element contains information supplemental or ancillary to the other data specified in the Supervision segment.

Example:

```
<j:ActivityCommentText>Probation revoked.
</j:ActivityCommentText>
```

#### **Corrections or Jail Case Number (<rap:SupervisionRecordID>):**

This element contains the case number assigned by the jail, prison, probation officer, or the like. This element is also called the Originating Case Agency number (OCA).

Example:

```
<rap:SupervisionRecordID>
  <j:ID>1998AF002356</j:ID>
</rap:SupervisionRecordID>
```

#### **Court Case Number (<rap:CourtRecordID>):**

This element contains the case number assigned by the court. This element is also called the Originating Case Agency number (OCA).

Example:

```
<rap:CourtRecordID>
  <j:ID>1998AF002356</j:ID>
</rap:CourtRecordID>
```

#### **Subject Under Supervision (<j:SupervisionSubject>):**

**GJXDM:** A person who is being supervised.

This group element contains the name and local identifier for the person in custody or under supervision.

**Subject Name (<j:PersonName>):**

**GJXDM:** A name by which a person is known.

This element should include all names for the record subject, including “also known as” (AKA) alias names. Use the <j:PersonFullName> subelement.

Example:

```
<j:PersonName>
  <j:PersonFullName>Mitch Doherty</j:PersonFullName>
</j:PersonName>
<j:PersonName>
  <j:PersonFullName>Michael Doherty</j:PersonFullName>
</j:PersonName>
```

**Corrections Identification Number (<j:SubjectID>):**

**GJXDM:** An assigned number or string that identifies a subject..

The element contains a correctional subject’s identification number for a given state. The NCIC state two-letter code must be specified in the IDIssuingAuthorityText attribute. The type attribute is required. This number is often like the SID number assigned at the state identification bureau or the FNU assigned at the federal level.

Example:

```
<j:SubjectID>
  <j:ID>48204395</j:ID>
</j:SubjectID>
```

**Supervision Agency (<j:SupervisionAgency>):**

**GJXDM:** An agency responsible for supervising a subject.

This element contains information about the correctional facility, jail, probation or parole agency responsible for the supervising the subject. One or both of the subelements <j:OrganizationName> or <j:OrganizationORIID> must be present.

Example:

```
<j:SupervisionAgency>
  <j:OrganizationName>Portage County Jail
  </j:OrganizationName>
  <j:OrganizationORIID>
  <j:ID>WI014400Y</j:ID>
  </j:OrganizationORIID>
</j:SupervisionAgency>
```

**Action Literal (<j:SupervisionCustodyStatus>):**

**GJXDM:** A status of the custody of a person under supervision.

This element contains a description of the corrections action (such as, receipt, release, transfer, escape, etc.).

Example:

```
<j:SupervisionCustodyStatus>incarcerated
  </j:SupervisionCustodyStatus>
```

### **Supervision Release Date (<j:SupervisionRelease>):**

**GJXDM:** A complete and unrestricted release of a subject from supervision.

Use the <j:ActivityDate> subelement to report the actual or planned date on which this person will be or has been released. The format for the date is CCYY[-MM[-DD]].

Example:

```
<j:SupervisionRelease>
  <j:ActivityDate>1998-06-30</j:ActivityDate>
</j:SupervisionRelease>
```

## **3.4 Agency Index (<j:Organization>)**

The Index element contains Name, ORI, and contact information for an Agency identified in the rap sheet file. This element can repeat as many times as necessary.

### **Agency (<j:Organization>):**

**GJXDM:** Details about a unit which conducts some sort of business or operations.

This element contains information about an agency itemized in this index. If the <j:OrganizationName> element is not transmitted, then the <j:OrganizationORIID> element is mandatory. It is expected, however, that this section of the rapsheet would contain *current*, detailed information about all of the agencies referenced elsewhere. It is understood that systems generating rapsheet data may have to obtain current agency information from a source other than the criminal history record system.

### **Agency Name (<j:OrganizationName>):**

**GJXDM:** A name of an organization.

The entry of an ORI number in the <j:OrganizatonORIID> element is preferred over the entry of the agency name or any other optional field; however, in the situation that an ORI is not available for entry, the agency name is mandatory.

Example:

```
<j:OrganizationName>Lodi Police Department
</j:OrganizationName>
```

### **Agency ORI (<j:OrganizationORIID>):**

**GJXDM:** A unique identifier assigned to a justice-related organization by the federal government. Sometimes referred to as ORI, ORI code, Originating Agency Code.

An ORI is a nine-character “ORiginating agency Identifier” assigned to an agency by the FBI.

Example:

```
<j:OrganizationORIID>  
  <j:ID>WI0111000</j:ID>  
</j:OrganizationORIID>
```

**Agency Address (<j:OrganizationLocation>):**

**GJXDM:** A location of an organization.

Report the address of the agency.

**Address (<j:LocationAddress>):**

**GJXDM:** Details about an address for a location, e.g., a postal address.

**Address Text (<j:AddressFullText>):**

**GJXDM:** A complete address.

This element may be used when the address components cannot be broken down into the GJXDM components. This element simply contains an unformatted text string containing the location's address.

**Street information (<j:LocationStreet>)**

**GJXDM:** Details about a street.

**Street information (<j:StreetFullText>):**

**GJXDM:** A complete street reference, e.g., "123 Main Street NW".

This element would contain an entire address line, like 1565 N. Park Place. This element may be repeated if necessary.

**Apartment or Suite information**

**(<j:LocationSecondaryUnitText>):**

**GJXDM:** A piece of information used to identify a particular unit within a specific location. Examples of this could be an apartment number or a suite number.

This element holds an optional apartment or a suite number for this location.

**Post Office Box information**

**(<j:AddressMailDeliveryUnitText>):**

**GJXDM:** A complete Post Office box number or other mail delivery point for a location, e.g., "PO BOX 12345"

This element holds an optional post office box number for this location.

**City (<j:LocationCityName>):**

**GJXDM:** A name of a city or town.

This element contains the city for this location.



**County (<j:LocationCountyName>):**

**GJXDM:** A code identifying a county, parish, or vicinage.

This element contains the county for this location.

**State (<j:LocationStateName>):**

**GJXDM:** A name of a state, commonwealth, province, or other subregion of a country.

This element contains the name of the state.

**Zip code (<j:LocationPostalCodeID>):**

**GJXDM:** A zip code or postal code.

This element contains the 5 or 9 digit zipcode or foreign postal code for this location. The actual value of the number is reported in a <j:ID> subelement.

**Country (<j:LocationCountryName>):**

**GJXDM:** A name of a country.

This element contains the country for this location.

**Agency Contact (<j:LocationContactInformation>):**

**GJXDM:** Details about contact information for a location.

This element contains contact information.

**Contact Person**

**(<j:ContactInformationDescriptionText>):**

**GJXDM:** A general description of contact information.

This element contains the name or title of a contact person.

**Voice Phone (<j:ContactTelephoneNumber>):**

**GJXDM:** A telephone number of a person or organization.

**Number (<j:TelephoneNumberFullID>):**

**GJXDM:** A full length voice telephone identifier.

**Fax (<j:ContactFacsimileNumber>):**

**GJXDM:** A facsimile number of a person or organization. Sometimes referred to as a FAX or fax number.

**Number (<j:TelephoneNumberFullID>):**

**GJXDM:** A full length telephone identifier.

This element contains the full fax phone number.

**Email (<j:ContactEmailID>):**

**GJXDM:** An email address of a person or organization..

The <j:ID> subelement contains the contact person's email address.

Example:

```
<j:OrganizationLocation>
  <j:LocationAddress>
    <j:LocationStreet>
      <j:StreetFullText>1565 N Park
Place</j:StreetFullText>
    </j:LocationStreet>
    <j:LocationCityName>Hamilton</j:LocationCityName>
    <j:LocationStateName>NJ</j:LocationStateName>
    <j:LocationPostalCodeID>08610</j:LocationPostalCodeID>
  </j:LocationAddress>
  <j:LocationContactInformation>
    <j:ContactInformationDescriptionText>George
      </j:ContactInformationDescriptionText>
    <j:ContactTelephoneNumber>
      <j:TelephoneNumberFullID>4859304869
        </j:TelephoneNumberFullID>
    </j:ContactTelephoneNumber>
    <j:ContactFacsimileNumber>
      <j:TelephoneNumberFullID>4859304870
        </j:TelephoneNumberFullID>
    </j:ContactFacsimileNumber>
    <j:ContactEmailID>george@lodiPD.gov</j:ContactEmailID>
  </j:LocationContactInformation>
</j:OrganizationLocation>
```

**Agency Abbreviation (<j:OrganizationAbbreviationText>):**

**GJXDM:** An abbreviation, acronym, or code for an organization name, e.g., FBI, NCIC.

Use this element to report agency name abbreviations or acronyms.

Example:

```
<j:OrganizationAbbreviationText>LOPD
  </j:OrganizationAbbreviationText>
```

**Agency Unit (<j:OrganizationUnitName>):**

**GJXDM** A name of a high-level division of an organization, e.g., department, agency.

Use this element to report a particular agency unit or department to contact.

Example:

```
<j:OrganizationUnitName>Records</j:OrganizationUnitName>
```

## **4.0 *Rap Sheet Definition***

This section contains the full rap sheet definition. Section 4.1 gives the structure in table form. Section 4.2 gives a formal XML Schema definition of the root `<rap:Rapsheet>` element and all its child elements.

### **4.1 Detailed Structure**

Comment	ELEMENT NAME	Occurs		Attribute Usage	ANSI/NIST
		MIN	MAX		
rap:RapSheet		1	1	@reportedDate,@xmlns,@version	
	rap:Introduction	1	1		
return the	rap:RapSheetRequest	1	1	@id	
requestor's	j:PersonFBIID	0	1		
search parameters	j:ID	1	1		
	j:PersonStateID	0	1		
	j:ID	1	1		
	j:PersonSSNID	0	unbounded		
	j:ID	1	1		
	j:PersonDriverLicenseID	0	unbounded		
	j:ID	1	1		
	j:PersonOtherID	0	unbounded		
	j:IDTypeText	1	1		
	j:ID	1	1		
	j:IDIssuingAuthorityText	0	1		
	j:PersonFullName	0	unbounded		
	j:PersonSexText	0	unbounded		
	j:PersonRaceText	0	unbounded		
	j:PersonBirthDate	0	unbounded		
	rap:PurposeCode	1	1		
	rap:Attention	1	1		
	rap:Caveat	0	unbounded		2.006 ATN
qualify your	j:CaveatText	1	1		
response	j:CaveatReferenceDate	0	1		

	j:CaveatIssuingAuthorityName	1	1		
	rap:ControlData	0	unbounded	@typeText, @issuingAuthorityName	
IDENTIFICATION	j:Subject	1	1		
name	j:PersonName	1	unbounded		2.018 NAM
	j:PersonPrefixName	0	unbounded		
	j:PersonGivenName	0	1		
	j:PersonMiddleName	0	unbounded		
	j:PersonSurName	1	1		
	j:PersonSuffixName	0	unbounded		
alias	j:PersonAlternateName	0	unbounded		2.019 AKA
	j:PersonPrefixName	0	unbounded		
	j:PersonGivenName	0	1		
	j:PersonMiddleName	0	unbounded		
	j:PersonSurName	1	1		
	j:PersonSuffixName	0	unbounded		
residence	j:Residence	0	unbounded		
	j:LocationAddress	0	1	@reportedDate	2.041 RES
	j:AddressFullText	0	1		
	j:LocationStreet	0	1		
	j:StreetFullText	1	1		
	j:LocationSecondaryUnitText	0	1		
	j:AddressMailDeliveryUnitText	0	1		
	j:LocationCityName	0	1		
	j:LocationCountyName	0	1		
	j:LocationStateName	0	1		
	j:LocationPostalCodeID	0	1		
	j:ID	1	1		
	j:LocationCountryName	0	1		
	j:LocationContactInformation	0	1		

employer
occupation
birthdate
birthplace
deceased
sex offender
iffs disqualified

	j:ContactTelephoneNumber	1	unbounded
	j:TelephoneNumberFullID	1	1
	j:Employment	0	unbounded
	j:EmploymentEmployerName	1	1
	j:EmploymentOccupationText	1	1
	j:EmploymentLocation	0	1
	j:LocationAddress	1	1
	j:AddressFullText	0	1
	j:LocationStreet	0	1
	j:StreetFullText	1	1
	j:LocationSecondaryUnitText	0	1
	j:AddressMailDeliveryUnitText	0	1
	j:LocationCityName	0	1
	j:LocationCountyName	0	1
	j:LocationStateName	0	1
	j:LocationPostalCodeID	0	1
	j:ID	1	1
	j:LocationCountryName	0	1
	j:EmploymentContactInformation	0	1
	j:ContactTelephoneNumber	1	unbounded
	j:TelephoneNumberFullID	1	1
	j:PersonBirthDate	0	unbounded
	j:PersonBirthLocation	0	unbounded
	j:LocationName	1	1
	j:PersonDeathDate	0	unbounded
	j:PersonLivingIndicator	0	unbounded
	j:PersonRegisteredOffenderIndicator	0	unbounded
	j:PersonFirearmSalesDisqualifiedIndicator	0	unbounded

2.039 EAD  
2.040 OCP

2.022 DOB  
2.020 POB

@reportingOrganizationT  
ext

ssn
drivers license
FBI number
SID
MNU
height
weight
eye color
hair color
sex
race
skin tone
scars marks tattoos

j:PersonAssignedIDDetails	1	1	
j:PersonSSNID	0 unbounded		
j:ID	1	1	2.016 SOC
j:PersonDriverLicenseID	0 unbounded		
j:ID	1	1	
j:PersonFBIID	1	1	
j:ID	1	1	2.014 FBI
j:PersonStateID	1 unbounded		
j:ID	1	1	2.015 SID
j:PersonOtherID	0 unbounded		
j:IDTypeText	1	1	
j:ID	1	1	2.017 MNU
j:IDIssuingAuthorityText	1	1	
j:PersonPhysicalDetails	1	1	
j:PersonHeightMeasure	0 unbounded	@personHeightUnitCode,	
		@r	2.027 HGT
j:PersonWeightMeasure	0 unbounded	@personWeightUnitCode	
		,@r	2.029 WGT
j:PersonEyeColorText	0 unbounded	@reportedDate	2.031 EYE
j:PersonHairColorText	0 unbounded	@reportedDate	2.032 HAI
j:PersonSexText	1 unbounded		2.024 SEX
j:PersonRaceText	1 unbounded		2.025 RAC
j:PersonSkinToneText	0 unbounded		
j:PersonPhysicalFeature	0 unbounded		
j:PhysicalFeatureCategoryText	0	1	
j:PhysicalFeatureTypeCode	0	1	2.026 SMT
j:PhysicalFeatureDescriptionText	0	1	
j:PhysicalFeatureImage	0 unbounded		
j:BinaryObject.Base64	0	1	

ethnicity
citizenship
marital status
religion
mugshot

	j:BinaryFormatText	0	1
	j:BinarySizeValue	0	1
	j:BinaryTypeText	0	1
	j:BinaryDescriptionText	0	1
	j:BinaryReferenceID	0	1
	j:ID	1	1
	j:BinaryCaptureDate	0	1
	j:BinaryCaptureOrganization	1	1
	j:OrganizationName	1	1
	j:OrganizationORIID	0	1
	j:ID	1	1
	j:PersonSocialDetails	0	1
	j:PersonEthnicityText	0	unbounded
	j:PersonCitizenshipText	0	unbounded
	j:PersonMaritalStatusText	0	unbounded
	j:PersonReligionText	0	unbounded
	j:PersonBiometricDetails	0	1
	j:PersonDigitalImage	0	unbounded
	j:BinaryObject.Base64	0	1
	j:BinaryFormatText	0	1
	j:BinarySizeValue	0	1
	j:BinaryTypeText	0	1
	j:BinaryDescriptionText	0	1
	j:BinaryReferenceID	0	1
	j:ID	1	1
	j:BinaryCaptureDate	0	1
	j:BinaryCaptureOrganization	1	1
	j:OrganizationName	1	1
	j:OrganizationORIID	0	1

@reportedDate  
@reportedDate  
@reportedDate

2.021 CTZ

2.036 PHT





	j:DNALocusTypeText	1	1	
	j:DNALocusValue	1	2	
	j:PersonPalmPrint	0	unbounded	
palm print image	j:BiometricImage	1	unbounded	
	j:BinaryObject.Base64	0	1	
	j:BinaryFormatText	0	1	
	j:BinarySizeValue	0	1	
	j:BinaryTypeText	0	1	
	j:BinaryDescriptionText	0	1	
	j:BinaryReferenceID	0	1	
	j:ID	1	1	
	j:BinaryCaptureDate	0	1	
	j:BinaryCaptureOrganization	1	1	2.035 PPA
	j:OrganizationName	1	1	
	j:OrganizationORIID	0	1	
	j:ID	1	1	
	j:PersonMedicalDetails	0	1	
	j:PersonMedicalCondition (@reportedDate)	1	unbounded	
	j:MedicalConditionText	1	1	
caution	j:SubjectCautionInformationCaveat	0	unbounded	
	j:SubjectOffenderNoticeText	0	unbounded	@reportedDate, @reportin gOr
HISTORY	rap:Cycle	0	unbounded	2.056 ICO
	rap:CycleEarliestDate	1	1	
	rap:CycleTrackingID	0	unbounded	
	j:ID	1	1	
	j:Incident	0	unbounded	
date of offense	j:ActivityDate	1	1	2.048 DOO
	j:Arrest	0	unbounded	

adult/juvenile arrest
date of arrest
arrest comment
arrest OCA
arrest subject name
local ID number
arrest/booking charge
charge comment
arrest felony/misdemeanor
counts
charges dropped

j:ActivityTypeText	0	1	
j:ActivityDate	1	1	2.087 TAA
j:ActivityCommentText	0	unbounded	2.045 DOA
j:ArrestAgencyRecordID	0	1	
j:ID	1	1	2.009 OCA
j:ArrestSubject	0	1	
j:PersonName	0	unbounded	
j:PersonFullName	1	1	
j:SubjectID	0	1	
j:ID	1	1	
j:ArrestCharge	1	unbounded	
j:ChargeID	0	1	
j:ID	1	1	
j:ChargeText	1	1	2.047 AOL
j:ChargeSequenceID	0	1	
j:ID	1	1	
j:ChargeTrackingID	0	1	
j:ID	1	1	
j:ChargeDescriptionText	0	unbounded	
j:ChargeClassification	1	1	
j:ChargeApplicabilityText	0	unbounded	
j:ChargeReducingFactorText	0	unbounded	
j:ChargeSpecialAllegationText	0	unbounded	
j:ChargeSeverityText	1	1	
j:ChargeCountQuantity	0	1	
j:ChargeDisposition	0	unbounded	@commentText
j:ChargeDispositionDate	1	1	
j:ChargeDispositionDescriptionText	1	1	
j:ChargeReporting	0	1	

offense code
arrest statute
arrest agency ORI
booking OCA
booking agency ORI
prosecution OCA

	j:ChargeNCICCode	1	unbounded
	j:ChargeStatute	0	unbounded
	j:StatuteCodeID	0	1
	j:ID	1	1
	j:StatuteText	1	0
	j:StatuteJurisdiction	0	1
	j:LocationStateName	1	1
	j:ArrestAgency	1	1
	j:OrganizationName	1	1
	j:OrganizationORIID	0	1
	j:ID	1	1
j:Booking		0	unbounded
	j:BookingAgencyRecordID	0	1
	j:ID	1	1
	j:BookingAgency	1	1
	j:OrganizationName	1	1
	j:OrganizationORIID	0	1
	j:ID	1	1
rap:Prosecution		0	unbounded
	j:ActivityDate	0	1
	j:ActivityCommentText	0	unbounded
	rap:ProsecutionAgencyRecordID	0	1
	j:ID	1	1
	rap:ProsecutionSubject	0	1
	j:PersonName	0	unbounded
	j:PersonFullName	1	1
	j:SubjectID	0	1
	j:ID	1	1
	j:ProsecutionCharge	1	unbounded

2.010 CIN

prosecution charge
decline to prosecute
prosecution agency ORI

j:ChargeID	0	1
j:ID	1	1
j:ChargeText	1	1
j:ChargeSequenceID	0	1
j:ID	1	1
j:ChargeTrackingID	0	1
j:ID	1	1
j:ChargeDescriptionText	0	unbounded
j:ChargeClassification	1	1
j:ChargeApplicabilityText	0	unbounded
j:ChargeReducingFactorText	0	unbounded
j:ChargeSpecialAllegationText	0	unbounded
j:ChargeSeverityText	1	1
j:ChargeCountQuantity	0	1
j:ChargeDisposition	0	unbounded @commentText
j:ChargeDispositionDate	1	1
j:ChargeDispositionDescriptionText	1	1
j:ChargeReporting	0	1
j:ChargeNCICCode	1	unbounded
j:ChargeStatute	0	unbounded
j:StatuteCodeID	0	1
j:ID	1	1
j:StatuteText	1	0
j:StatuteJurisdiction	0	1
j:LocationStateName	1	1
rap:ProsecutionAgency	1	1
j:OrganizationName	1	1
j:OrganizationORIID	0	1
j:ID	1	1

court disposition date
court OCA
court charge
convicted, acquitted . . .

rap:CourtAction	0	unbounded
j:ActivityDate	1	1
j:ActivityCommentText	0	unbounded
rap:CourtRecordID	0	1
j:ID	1	1
rap:CourtSubject	0	1
j:PersonName	0	unbounded
j:PersonFullName	1	1
j:SubjectID	0	1
j:ID	1	1
j:CourtCharge	1	unbounded
j:ChargeID	0	1
j:ID	1	1
j:ChargeText	1	1
j:ChargeSequenceID	0	1
j:ID	1	1
j:ChargeTrackingID	0	1
j:ID	1	1
j:ChargeDescriptionText	0	unbounded
j:ChargeClassification	1	1
j:ChargeApplicabilityText	0	unbounded
j:ChargeReducingFactorText	0	unbounded
j:ChargeSpecialAllegationText	0	unbounded
j:ChargeSeverityText	1	1
j:ChargeCountQuantity	0	1
j:ChargeDisposition	0	unbounded @commentText
j:ChargeDispositionDate	1	1
j:ChargeDispositionDescriptionText	1	1
j:ChargeReporting	0	1

2.050 CDD

2.051 CSL

court ORI
sentence

	j:ChargeNCICCode	1	unbounded
	j:ChargeStatute	0	unbounded
	j:StatuteCodeID	0	1
	j:ID	1	1
	j:StatuteText	1	0
	j:StatuteJurisdiction	0	1
	j:LocationStateName	1	1
j:Co		1	1
	j:OrganizationName	1	1
	j:OrganizationORIID	0	1
	j:ID	1	1
rap:Sentencing		0	unbounded
	j:ActivityDate	1	1
	j:ActivityCommentText	0	unbounded
	rap:CourtRecordID	0	1
	j:ID	1	1
j:Se		1	unbounded
	j:SentenceCharge	1	unbounded
	j:ChargeID	0	1
	j:ID	1	1
	j:ChargeText	1	1
	j:ChargeSequenceID	0	1
	j:ID	1	1
	j:ChargeTrackingID	0	1
	j:ID	1	1
	j:SentenceDescriptionText	0	unbounded
j:Co		0	1
	j:OrganizationName	1	1
	j:OrganizationORIID	0	1





	j:ChargeDispositionDate	1	1
	j:ChargeDispositionDescriptionText	1	1
	j:ChargeReporting	0	1
	j:ChargeNCICCode	1	unbounded
	j:ChargeStatute	0	unbounded
	j:StatuteCodeID	0	1
	j:ID	1	1
	j:StatuteText	1	0
	j:StatuteJurisdiction	0	1
	j:LocationStateName	1	1
	j:SupervisionAgency	1	1
	j:OrganizationName	1	1
	j:OrganizationORIID	0	1
corrections agency ORI	j:ID	1	1
jail, probation, etc.	j:SupervisionCustodyStatus	1	unbounded
	j:SupervisionRelease	0	1
corrections end date	j:ActivityDate	1	1
	j:Organization	1	unbounded
	j:OrganizationName	1	1
	j:OrganizationORIID	0	1
INDEX of Agencies	j:ID	1	1
and detailed	j:OrganizationLocation	0	1
address and contact	j:LocationAddress	0	1
information.	j:AddressFullText	0	1
	j:LocationStreet	0	1
	j:StreetFullText	1	1
	j:LocationSecondaryUnitText	0	1
	j:AddressMailDeliveryUnitText	0	1
	j:LocationCityName	0	1

2.055 SLE







## 4.2 XML Schemas

The following table contains the XML schemas for the rap sheet. The schema defines the structure for the XML rap sheet document. The element hierarchy is defined along with information regarding whether the field is mandatory and maximum number of occurrences permitted. This schema is intended for use by a validating parser to validate the structure of an XML rap sheet document.

The example in Section 5 was validated against the following schema using the Xerces 1.3.0 validating parser available from the Apache group ([www.apache.org](http://www.apache.org)).

### 4.2.1 Document/Extension Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:rap-code="http://www.nlets.org/rapsheet/proxy/codes/1.0"
xmlns:j="http://www.it.ojp.gov/jxdm/3.0" xmlns:rap="http://www.nlets.org/rapsheet/3.0.0"
targetNamespace="http://www.nlets.org/rapsheet/3.0.0" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:import namespace="http://www.it.ojp.gov/jxdm/3.0" schemaLocation="jxdm/3.0/jxdm.xsd"/>
  <xsd:import namespace="http://www.nlets.org/rapsheet/proxy/codes/1.0" schemaLocation="RapSheetCodes-proxy.xsd"/>
  <xsd:element name="RapSheet" type="rap:RapSheetType"/>
  <xsd:element name="Attention" type="xsd:string"/>
  <xsd:element name="ControlData" type="rap:StringWithType"/>
  <xsd:element name="Cycle" type="rap:CycleType"/>
  <xsd:element name="CycleEarliestDate" type="xsd:date"/>
  <xsd:element name="CycleTrackingID" type="j:IDType"/>
  <xsd:element name="Introduction" type="rap:IntroductionType"/>
  <xsd:element name="Caveat" type="rap:CaveatType"/>
  <xsd:element name="Prosecution" type="rap:ProsecutionType"/>
  <xsd:element name="ProsecutionAgencyRecordID" type="j:IDType"/>
  <xsd:element name="ProsecutionSubject" type="j:SubjectType"/>
  <xsd:element name="ProsecutionAgency" type="j:OrganizationType"/>
  <xsd:element name="CourtAction" type="rap:CourtActionType"/>
  <xsd:element name="CourtRecordID" type="j:IDType"/>
```

```

<xsd:element name="CourtSubject" type="j:SubjectType"/>
<xsd:element name="Sentencing" type="rap:SentencingType"/>
<xsd:element name="Supervision" type="rap:SupervisionType"/>
<xsd:element name="SupervisionCharge" type="rap:SupervisionChargeType"/>
<xsd:element name="SupervisionAgencyRecordID" type="j:IDType"/>
<xsd:element name="PurposeCode" type="rap-code:PurposeCodeType"/>
<xsd:element name="RapSheetRequest" type="rap:RapSheetRequest"/>
<xsd:attribute name="typeText" type="xsd:string"/>
<!-- definition of complex types -->
<xsd:complexType name="RapSheetType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="rap:Introduction"/>
        <xsd:element ref="j:Subject"/>
        <xsd:element ref="rap:Cycle" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:Organization" maxOccurs="unbounded"/>
      </xsd:sequence>
      <xsd:attribute name="version" type="xsd:string" use="optional"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="IntroductionType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="rap:RapSheetRequest"/>
        <xsd:element ref="rap:Caveat" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="rap:ControlData" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CycleType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="rap:CycleEarliestDate"/>
        <xsd:element ref="rap:CycleTrackingID" minOccurs="0"/>
        <xsd:element ref="j:Incident" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:Arrest" minOccurs="0"/>
        <xsd:element ref="j:Booking" minOccurs="0"/>
        <xsd:element ref="rap:Prosecution" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="rap:CourtAction" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="rap:Sentencing" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="rap:Supervision" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="RapSheetRequest">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:PersonFBIID" minOccurs="0"/>
        <xsd:element ref="j:PersonStateID" minOccurs="0"/>
        <xsd:element ref="j:PersonSSNID" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:PersonDriverLicenseID" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:PersonOtherID" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:PersonFullName" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:PersonSexText" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:PersonRaceText" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

        <xsd:element ref="j:PersonBirthDate" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="rap:PurposeCode"/>
        <xsd:element ref="rap:Attention"/>
    </xsd:sequence>
    <xsd:attribute name="id" type="xsd:string" use="optional"/>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CaveatType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:CaveatText"/>
                <xsd:element ref="j:CaveatReferenceDate" minOccurs="0"/>
                <xsd:element ref="j:CaveatIssuingAuthorityName"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ProsecutionType">
    <xsd:complexContent>
        <xsd:extension base="j:ActivityType">
            <xsd:sequence>
                <xsd:element ref="rap:ProsecutionAgencyRecordID" minOccurs="0"/>
                <xsd:element ref="rap:ProsecutionSubject" minOccurs="0"/>
                <xsd:element ref="j:ProsecutionCharge" maxOccurs="unbounded"/>
                <xsd:element ref="rap:ProsecutionAgency"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```



```

<xsd:complexType name="CourtActionType">
  <xsd:complexContent>
    <xsd:extension base="j:ActivityType">
      <xsd:sequence>
        <xsd:element ref="rap:CourtRecordID" minOccurs="0"/>
        <xsd:element ref="rap:CourtSubject" minOccurs="0"/>
        <xsd:element ref="j:CourtCharge" maxOccurs="unbounded"/>
        <xsd:element ref="j:Court"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SentencingType">
  <xsd:complexContent>
    <xsd:extension base="j:ActivityType">
      <xsd:sequence>
        <xsd:element ref="rap:CourtRecordID" minOccurs="0"/>
        <xsd:element ref="j:Sentence" maxOccurs="unbounded"/>
        <xsd:element ref="j:Court" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SupervisionType">
  <xsd:complexContent>
    <xsd:extension base="j:ActivityType">
      <xsd:sequence>
        <xsd:element ref="rap:SupervisionAgencyRecordID" minOccurs="0"/>
        <xsd:element ref="rap:CourtRecordID" minOccurs="0"/>
        <xsd:element ref="j:SupervisionSubject" minOccurs="0"/>
        <xsd:element ref="rap:SupervisionCharge" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

        <xsd:element ref="j:SupervisionAgency"/>
        <xsd:element ref="j:SupervisionCustodyStatus" maxOccurs="unbounded"/>
        <xsd:element ref="j:SupervisionRelease" minOccurs="0"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SupervisionChargeType">
    <xsd:complexContent>
        <xsd:extension base="j:ChargeType"/>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="StringWithType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:attribute ref="rap:typeText"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
</xsd:schema>

```

#### 4.2.2 GJXDM Constraint Schema

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:j="http://www.it.ojp.gov/jxdm/3.0" xmlns:rap-code="http://www.nlets.org/rapsheet/proxy/codes/1.0"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:j-ncic="http://www.it.ojp.gov/jxdm/3.0/proxy/ncic_2000/1.0" xmlns:j-
xsd="http://www.it.ojp.gov/jxdm/3.0/proxy/xsd/1.0" targetNamespace="http://www.it.ojp.gov/jxdm/3.0"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xsd:import namespace="http://www.it.ojp.gov/jxdm/3.0/proxy/xsd/1.0" schemaLocation="proxy/xsd/1.0/xsd.xsd"/>
    <xsd:import namespace="http://www.it.ojp.gov/jxdm/3.0/proxy/ncic_2000/1.0"
schemaLocation="proxy/ncic_2000/1.0/ncic_2000.xsd"/>

```

```

<xsd:import namespace="http://www.nlets.org/rapsheet/proxy/codes/1.0" schemaLocation="../../RapSheetCodes-proxy.xsd"/>
<xsd:attributeGroup name="SuperTypeMetadata">
  <xsd:attribute ref="j:reportingOrganizationText" use="optional"/>
  <xsd:attribute ref="j:reportedDate" use="optional"/>
  <xsd:attribute ref="j:commentText" use="optional"/>
</xsd:attributeGroup>
<xsd:complexType name="ActivityType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:ActivityTypeText" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:ActivityDate" minOccurs="0"/>
        <xsd:element ref="j:ActivityCommentText" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="AddressType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:AddressFullText" minOccurs="0"/>
        <xsd:element ref="j:LocationStreet" minOccurs="0"/>
        <xsd:element ref="j:LocationSecondaryUnitText" minOccurs="0"/>
        <xsd:element ref="j:AddressMailDeliveryUnitText" minOccurs="0"/>
        <xsd:element ref="j:LocationCityName" minOccurs="0"/>
        <xsd:element ref="j:LocationCountyName" minOccurs="0"/>
        <xsd:element ref="j:LocationStateName" minOccurs="0"/>
        <xsd:element ref="j:LocationPostalCodeID" minOccurs="0"/>
        <xsd:element ref="j:LocationCountryName" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ArrestType">
    <xsd:complexContent>
        <xsd:extension base="j:ActivityType">
            <xsd:sequence>
                <xsd:element ref="j:ArrestAgencyRecordID" minOccurs="0"/>
                <xsd:element ref="j:ArrestSubject" minOccurs="0"/>
                <xsd:element ref="j:ArrestCharge" maxOccurs="unbounded"/>
                <xsd:element ref="j:ArrestAgency"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="BinaryType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:BinaryObject.Base64" minOccurs="0"/>
                <xsd:element ref="j:BinaryFormatText" minOccurs="0"/>
                <xsd:element ref="j:BinarySizeValue" minOccurs="0"/>
                <xsd:element ref="j:BinaryTypeText" minOccurs="0"/>
                <xsd:element ref="j:BinaryDescriptionText" minOccurs="0"/>
                <xsd:element ref="j:BinaryReferenceID" minOccurs="0"/>
                <xsd:element ref="j:BinaryCaptureDate" minOccurs="0"/>
                <xsd:element ref="j:BinaryCaptureOrganization"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

<xsd:complexType name="BiometricType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:BiometricValueText" minOccurs="0"/>
        <xsd:element ref="j:BiometricEncodingMethodText" minOccurs="0"/>
        <xsd:element ref="j:BiometricDescriptionText" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:BiometricImage" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:BiometricCaptureOrganization" minOccurs="0"/>
        <xsd:element ref="j:BiometricCaptureDate" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="BookingType">
  <xsd:complexContent>
    <xsd:extension base="j:ActivityType">
      <xsd:sequence>
        <xsd:element ref="j:BookingAgencyRecordID" minOccurs="0"/>
        <xsd:element ref="j:BookingAgency"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ChargeClassificationType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:ChargeApplicabilityText" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:ChargeReducingFactorText" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:ChargeSpecialAllegationText" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

                <xsd:element ref="j:ChargeSeverityText"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ChargeDispositionType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:ChargeDispositionDate"/>
                <xsd:element ref="j:ChargeDispositionDescriptionText"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ChargeReportingType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:ChargeNCICCode" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ChargeType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:ChargeID" minOccurs="0"/>
                <xsd:element ref="j:ChargeText"/>
                <xsd:element ref="j:ChargeSequenceID" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        <xsd:element ref="j:ChargeTrackingID" minOccurs="0"/>
        <xsd:element ref="j:ChargeDescriptionText" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:ChargeClassification" minOccurs="0"/>
        <xsd:element ref="j:ChargeCountQuantity" minOccurs="0"/>
        <xsd:element ref="j:ChargeDisposition" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:ChargeReporting" minOccurs="0"/>
        <xsd:element ref="j:ChargeStatute" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ContactInformationType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:ContactInformationDescriptionText" minOccurs="0"/>
                <xsd:element ref="j:ContactTelephoneNumber" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:ContactFacsimileNumber" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:ContactEmailID" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="CourtType">
    <xsd:complexContent>
        <xsd:extension base="j:OrganizationType"/>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="DNALocusType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">

```

```

                <xsd:sequence>
                    <xsd:element ref="j:DNALocusTypeText"/>
                    <xsd:element ref="j:DNALocusValue" maxOccurs="2"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="DNAType">
        <xsd:complexContent>
            <xsd:extension base="j:BiometricType">
                <xsd:sequence>
                    <xsd:element ref="j:DNALocus" minOccurs="0" maxOccurs="14"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="DispositionType">
        <xsd:complexContent>
            <xsd:extension base="j:SuperType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="DrivingJurisdictionAuthorityIDType">
        <xsd:complexContent>
            <xsd:extension base="j:IDType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="EmploymentType">
        <xsd:complexContent>
            <xsd:extension base="j:SuperType">
                <xsd:sequence>
                    <xsd:element ref="j:EmploymentEmployerName" minOccurs="0"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

```



```

        <xsd:element ref="j:EmploymentOccupationText" minOccurs="0"/>
        <xsd:element ref="j:EmploymentLocation" minOccurs="0"/>
        <xsd:element ref="j:EmploymentContactInformation" minOccurs="0"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="FingerprintSetType">
    <xsd:complexContent>
        <xsd:extension base="j:BiometricType">
            <xsd:sequence>
                <xsd:element ref="j:Fingerprint" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="FingerprintType">
    <xsd:complexContent>
        <xsd:extension base="j:BiometricType">
            <xsd:sequence>
                <xsd:element ref="j:FingerprintFingerText" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="IDType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:ID"/>
                <xsd:element ref="j:IDTypeText" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

                <xsd:element ref="j:IDIssuingAuthorityText" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="IncidentType">
    <xsd:complexContent>
        <xsd:extension base="j:ActivityType"/>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ImageType">
    <xsd:complexContent>
        <xsd:extension base="j:BinaryType"/>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="JurisdictionType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:LocationStateName" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="LocationType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:LocationName" minOccurs="0"/>
                <xsd:element ref="j:LocationAddress" minOccurs="0"/>
                <xsd:element ref="j:LocationContactInformation" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="MeasureType">
    <xsd:simpleContent>
      <xsd:extension base="xsd:decimal">
        <xsd:attributeGroup ref="j:SuperTypeMetadata"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
  <xsd:complexType name="OrganizationType">
    <xsd:complexContent>
      <xsd:extension base="j:SuperType">
        <xsd:sequence>
          <xsd:element ref="j:OrganizationName"/>
          <xsd:element ref="j:OrganizationORIID" minOccurs="0"/>
          <xsd:element ref="j:OrganizationLocation" minOccurs="0"/>
          <xsd:element ref="j:OrganizationAbbreviationText" minOccurs="0"
maxOccurs="unbounded"/>
          <xsd:element ref="j:OrganizationUnitName" minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="PersonAssignedIDDetailsType">
    <xsd:complexContent>
      <xsd:extension base="j:SuperType">
        <xsd:sequence>
          <xsd:element ref="j:PersonSSNID" minOccurs="0" maxOccurs="unbounded"/>
          <xsd:element ref="j:PersonDriverLicenseID" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

```

        <xsd:element ref="j:PersonFBIID"/>
        <xsd:element ref="j:PersonStateID" maxOccurs="unbounded"/>
        <xsd:element ref="j:PersonOtherID" minOccurs="0" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonBiometricDetailsType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:PersonDigitalImage" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonBloodTypeText" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonFingerprintSet" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonDNA" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonPalmPrint" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonHeightMeasureType">
    <xsd:simpleContent>
        <xsd:extension base="j:MeasureType">
            <xsd:attribute ref="j:personHeightUnitCode" use="optional"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="PersonHeightUnitCodeType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="ncic"/>
    </xsd:restriction>

```

```

</xsd:simpleType>
<xsd:complexType name="PersonMedicalConditionType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:MedicalConditionText"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonMedicalDetailsType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:PersonMedicalCondition" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonNameTextType">
  <xsd:simpleContent>
    <xsd:extension base="j:TextType"/>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="PersonNameType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:PersonPrefixName" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:PersonGivenName" minOccurs="0"/>
        <xsd:element ref="j:PersonMiddleName" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

        <xsd:element ref="j:PersonSurName" minOccurs="0"/>
        <xsd:element ref="j:PersonSuffixName" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:PersonFullName" minOccurs="0"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonOtherIDType">
    <xsd:complexContent>
        <xsd:extension base="j:IDType"/>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonPhysicalDetailsType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:PersonHeightMeasure" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonWeightMeasure" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonEyeColorText" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonHairColorText" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonSexText" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonRaceText" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonSkinToneText" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonPhysicalFeature" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonSocialDetailsType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">

```

```

        <xsd:sequence>
            <xsd:element ref="j:PersonEthnicityText" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="j:PersonCitizenshipText" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="j:PersonMaritalStatusText" minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element ref="j:PersonReligionText" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:PersonName" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonAlternateName" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:Residence" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:Employment" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonBirthDate" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonBirthLocation" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonDeathDate" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonLivingIndicator" minOccurs="0" maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonRegisteredOffenderIndicator" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonFirearmSalesDisqualifiedIndicator" minOccurs="0"
maxOccurs="unbounded"/>
                <xsd:element ref="j:PersonAssignedIDDetails" minOccurs="0"/>
                <xsd:element ref="j:PersonPhysicalDetails" minOccurs="0"/>
                <xsd:element ref="j:PersonSocialDetails" minOccurs="0"/>
                <xsd:element ref="j:PersonBiometricDetails" minOccurs="0"/>
                <xsd:element ref="j:PersonMedicalDetails" minOccurs="0"/>
            </xsd:sequence>

```

```

        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="PersonWeightMeasureType">
    <xsd:simpleContent>
        <xsd:extension base="j:MeasureType">
            <xsd:attribute ref="j:personWeightUnitCode" use="optional"/>
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="PersonWeightUnitCodeType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="ncic"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="PhysicalFeatureType">
    <xsd:complexContent>
        <xsd:extension base="j:SuperType">
            <xsd:sequence>
                <xsd:element ref="j:PhysicalFeatureCategoryText" minOccurs="0"/>
                <xsd:element ref="j:PhysicalFeatureTypeCode" minOccurs="0"/>
                <xsd:element ref="j:PhysicalFeatureDescriptionText" minOccurs="0"/>
                <xsd:element ref="j:PhysicalFeatureImage" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="ReleaseType">
    <xsd:complexContent>
        <xsd:extension base="j:ActivityType"/>
    </xsd:complexContent>

```



```

</xsd:complexType>
<xsd:complexType name="ResidenceType">
  <xsd:complexContent>
    <xsd:extension base="j:LocationType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SentenceType">
  <xsd:complexContent>
    <xsd:extension base="j:ActivityType">
      <xsd:sequence>
        <xsd:element ref="j:SentenceCharge" maxOccurs="unbounded"/>
        <xsd:element ref="j:SentenceDescriptionText" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:SentenceIssuer.Organization" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SeverityLevelType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType"/>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="StatusType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:StatusText" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

<xsd:complexType name="StatuteType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:StatuteCodeID" minOccurs="0"/>
        <xsd:element ref="j:StatuteOffenseID" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:element ref="j:StatuteText" maxOccurs="unbounded"/>
        <xsd:element ref="j:StatuteJurisdiction" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="StreetType">
  <xsd:complexContent>
    <xsd:extension base="j:SuperType">
      <xsd:sequence>
        <xsd:element ref="j:StreetFullText"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:complexType name="SubjectType">
  <xsd:complexContent>
    <xsd:extension base="j:PersonType">
      <xsd:sequence>
        <xsd:element ref="j:SubjectID" minOccurs="0"/>
        <xsd:element ref="j:SubjectCautionInformationCaveat" minOccurs="0"
maxOccurs="unbounded"/>
        <xsd:element ref="j:SubjectOffenderNoticeCaveat" minOccurs="0" maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>

```

```

        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="SuperType">
        <xsd:attributeGroup ref="j:SuperTypeMetadata"/>
    </xsd:complexType>
    <xsd:complexType name="SupervisionType">
        <xsd:complexContent>
            <xsd:extension base="j:ActivityType">
                <xsd:sequence>
                    <xsd:element ref="j:SupervisionSubject" minOccurs="0"/>
                    <xsd:element ref="j:SupervisionAgency"/>
                    <xsd:element ref="j:SupervisionCustodyStatus" maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="TelephoneNumberType">
        <xsd:complexContent>
            <xsd:extension base="j:SuperType">
                <xsd:sequence>
                    <xsd:element ref="j:TelephoneNumberFullID"/>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="TextType">
        <xsd:simpleContent>
            <xsd:extension base="j-xsd:string"/>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:attribute name="id" type="xsd:ID"/>

```

```
<xsd:attribute name="ref" type="xsd:IDREF"/>
<xsd:attribute name="measureUnitText" type="xsd:string"/>
<xsd:attribute name="personWeightUnitCode" type="j:PersonWeightUnitCodeType"/>
<xsd:attribute name="personHeightUnitCode" type="j:PersonHeightUnitCodeType"/>
<xsd:attribute name="reportedDate" type="xsd:date"/>
<xsd:attribute name="reportingOrganizationText" type="xsd:string"/>
<xsd:attribute name="commentText" type="xsd:string"/>
<xsd:element name="ActivityCommentText" type="j:TextType"/>
<xsd:element name="ActivityDate" type="j-xsd:date"/>
<xsd:element name="ActivityTypeText" type="j:TextType"/>
<xsd:element name="AddressFullText" type="j:TextType"/>
<xsd:element name="AddressMailDeliveryUnitText" type="j:TextType"/>
<xsd:element name="Agency" type="j:OrganizationType"/>
<xsd:element name="Arrest" type="j:ArrestType"/>
<xsd:element name="ArrestAgency" type="j:OrganizationType"/>
<xsd:element name="ArrestAgencyRecordID" type="j:IDType"/>
<xsd:element name="ArrestCharge" type="j:ChargeType"/>
<xsd:element name="ArrestSubject" type="j:SubjectType"/>
<xsd:element name="BinaryCaptureDate" type="j-xsd:date"/>
<xsd:element name="BinaryCaptureOrganization" type="j:OrganizationType"/>
<xsd:element name="BinaryDescriptionText" type="j:TextType"/>
<xsd:element name="BinaryFormatText" type="j:TextType"/>
<xsd:element name="BinaryObject.Base64" type="j-xsd:base64Binary"/>
<xsd:element name="BinaryReferenceID" type="j:IDType"/>
<xsd:element name="BinarySizeValue" type="j-xsd:integer"/>
<xsd:element name="BinaryTypeText" type="j:TextType"/>
<xsd:element name="BiometricCaptureDate" type="j-xsd:date"/>
<xsd:element name="BiometricCaptureOrganization" type="j:OrganizationType"/>
<xsd:element name="BiometricDescriptionText" type="j:TextType"/>
<xsd:element name="BiometricEncodingMethodText" type="j:TextType"/>
<xsd:element name="BiometricImage" type="j:ImageType"/>
```

```
<xsd:element name="BiometricValueText" type="j:TextType"/>
<xsd:element name="Booking" type="j:BookingType"/>
<xsd:element name="BookingAgency" type="j:OrganizationType"/>
<xsd:element name="BookingAgencyRecordID" type="j:IDType"/>
<xsd:element name="CaveatIssuingAuthorityName" type="j:TextType"/>
<xsd:element name="CaveatReferenceDate" type="j-xsd:date"/>
<xsd:element name="CaveatText" type="j:TextType"/>
<xsd:element name="ChargeApplicabilityText" type="j:TextType"/>
<xsd:element name="ChargeCountQuantity" type="j-xsd:nonNegativeInteger"/>
<xsd:element name="ChargeClassification" type="j:ChargeClassificationType"/>
<xsd:element name="ChargeDescriptionText" type="j:TextType"/>
<xsd:element name="ChargeDisposition" type="j:ChargeDispositionType"/>
<xsd:element name="ChargeDispositionDate" type="j-xsd:date"/>
<xsd:element name="ChargeDispositionDescriptionText" type="j:TextType"/>
<xsd:element name="ChargeDispositionOtherText" type="j:TextType"/>
<xsd:element name="ChargeID" type="j:IDType"/>
<xsd:element name="ChargeNCICCode" type="j-ncic:OFFType"/>
<xsd:element name="ChargeReporting" type="j:ChargeReportingType"/>
<xsd:element name="ChargeReducingFactorText" type="j:TextType"/>
<xsd:element name="ChargeSequenceID" type="j:IDType"/>
<xsd:element name="ChargeSeverityText" type="j:TextType"/>
<xsd:element name="ChargeSpecialAllegationText" type="j:TextType"/>
<xsd:element name="ChargeStatute" type="j:StatuteType"/>
<xsd:element name="ChargeText" type="j:TextType"/>
<xsd:element name="ChargeTrackingID" type="j:IDType"/>
<xsd:element name="ContactEmailID" type="j:IDType"/>
<xsd:element name="ContactFacsimileNumber" type="j:TelephoneNumberType"/>
<xsd:element name="ContactInformationDescriptionText" type="j:TextType"/>
<xsd:element name="ContactTelephoneNumber" type="j:TelephoneNumberType"/>
<xsd:element name="CourtCharge" type="j:ChargeType"/>
<xsd:element name="Court" type="j:CourtType"/>
```

```
<xsd:element name="DNALocus" type="j:DNALocusType"/>
<xsd:element name="DNALocusTypeText" type="rap-code:DNALocusType"/>
<xsd:element name="DNALocusValue" type="j:TextType"/>
<xsd:element name="Employment" type="j:EmploymentType"/>
<xsd:element name="EmploymentContactInformation" type="j:ContactInformationType"/>
<xsd:element name="EmploymentEmployerName" type="j:TextType"/>
<xsd:element name="EmploymentLocation" type="j:LocationType"/>
<xsd:element name="EmploymentOccupationText" type="j:TextType"/>
<xsd:element name="Fingerprint" type="j:FingerprintType"/>
<xsd:element name="FingerprintFingerText" type="rap-code:FingerprintFingerType"/>
<xsd:element name="ID" type="j:TextType"/>
<xsd:element name="IDIssuingAuthorityText" type="j:TextType"/>
<xsd:element name="IDTypeText" type="rap-code:IDTypeType"/>
<xsd:element name="Incident" type="j:IncidentType"/>
<xsd:element name="LocationAddress" type="j:AddressType"/>
<xsd:element name="LocationCityName" type="j:TextType"/>
<xsd:element name="LocationContactInformation" type="j:ContactInformationType"/>
<xsd:element name="LocationCountryName" type="j:TextType"/>
<xsd:element name="LocationName" type="j:TextType"/>
<xsd:element name="LocationCountyName" type="j:TextType"/>
<xsd:element name="LocationPostalCodeID" type="j:IDType"/>
<xsd:element name="LocationSecondaryUnitText" type="j:TextType"/>
<xsd:element name="LocationStateName" type="j:TextType"/>
<xsd:element name="LocationStreet" type="j:StreetType"/>
<xsd:element name="MedicalConditionText" type="j:TextType"/>
<xsd:element name="Organization" type="j:OrganizationType"/>
<xsd:element name="OrganizationAbbreviationText" type="j:TextType"/>
<xsd:element name="OrganizationName" type="j:TextType"/>
<xsd:element name="OrganizationLocation" type="j:LocationType"/>
<xsd:element name="OrganizationORIID" type="j:IDType"/>
<xsd:element name="OrganizationUnitName" type="j:TextType"/>
```

```
<xsd:element name="PersonAlternateName" type="j:PersonNameType"/>
<xsd:element name="PersonAssignedIDDetails" type="j:PersonAssignedIDDetailsType"/>
<xsd:element name="PersonBirthDate" type="j-xsd:date"/>
<xsd:element name="PersonBirthLocation" type="j:LocationType"/>
<xsd:element name="PersonBiometricDetails" type="j:PersonBiometricDetailsType"/>
<xsd:element name="PersonBloodTypeText" type="rap-code:BloodTypeType"/>
<xsd:element name="PersonCitizenshipText" type="j:TextType"/>
<xsd:element name="PersonDNA" type="j:DNAType"/>
<xsd:element name="PersonDeathDate" type="j-xsd:date"/>
<xsd:element name="PersonDigitalImage" type="j:ImageType"/>
<xsd:element name="PersonDriverLicenseID" type="j:DrivingJurisdictionAuthorityIDType"/>
<xsd:element name="PersonEthnicityText" type="rap-code:EthnicityType"/>
<xsd:element name="PersonEyeColorText" type="rap-code:EyeColorType"/>
<xsd:element name="PersonFBIID" type="j:IDType"/>
<xsd:element name="PersonFingerprintSet" type="j:FingerprintSetType"/>
<xsd:element name="PersonFirearmSalesDisqualifiedIndicator" type="j-xsd:boolean"/>
<xsd:element name="PersonFullName" type="j:PersonNameTextType"/>
<xsd:element name="PersonGivenName" type="j:PersonNameTextType"/>
<xsd:element name="PersonHairColorText" type="rap-code:HairColorType"/>
<xsd:element name="PersonHeightMeasure" type="j:PersonHeightMeasureType"/>
<xsd:element name="PersonLivingIndicator" type="j-xsd:boolean"/>
<xsd:element name="PersonMaritalStatusText" type="rap-code:MaritalStatusType"/>
<xsd:element name="PersonMedicalCondition" type="j:PersonMedicalConditionType"/>
<xsd:element name="PersonMedicalDetails" type="j:PersonMedicalDetailsType"/>
<xsd:element name="PersonMiddleName" type="j:PersonNameTextType"/>
<xsd:element name="PersonName" type="j:PersonNameType"/>
<xsd:element name="PersonOtherID" type="j:PersonOtherIDType"/>
<xsd:element name="PersonPalmPrint" type="j:BiometricType"/>
<xsd:element name="PersonPhysicalDetails" type="j:PersonPhysicalDetailsType"/>
<xsd:element name="PersonPhysicalFeature" type="j:PhysicalFeatureType"/>
<xsd:element name="PersonPrefixName" type="j:TextType"/>
```

```
<xsd:element name="PersonRaceText" type="rap-code:RaceType"/>
<xsd:element name="PersonRegisteredOffenderIndicator" type="j-xsd:boolean"/>
<xsd:element name="PersonReligionText" type="j:TextType"/>
<xsd:element name="PersonSSNID" type="j:IDType"/>
<xsd:element name="PersonSexText" type="rap-code:SexType"/>
<xsd:element name="PersonSkinToneText" type="rap-code:SkinToneType"/>
<xsd:element name="PersonSocialDetails" type="j:PersonSocialDetailsType"/>
<xsd:element name="PersonStateID" type="j:IDType"/>
<xsd:element name="PersonSuffixName" type="j:TextType"/>
<xsd:element name="PersonSurName" type="j:PersonNameTextType"/>
<xsd:element name="PersonWeightMeasure" type="j:PersonWeightMeasureType"/>
<xsd:element name="PhysicalFeatureCategoryText" type="j:TextType"/>
<xsd:element name="PhysicalFeatureDescriptionText" type="j:TextType"/>
<xsd:element name="PhysicalFeatureImage" type="j:ImageType"/>
<xsd:element name="PhysicalFeatureTypeCode" type="j-ncic:SMTType"/>
<xsd:element name="ProsecutionCharge" type="j:ChargeType"/>
<xsd:element name="Residence" type="j:ResidenceType"/>
<xsd:element name="Sentence" type="j:SentenceType"/>
<xsd:element name="SentenceCharge" type="j:ChargeType"/>
<xsd:element name="SentenceDescriptionText" type="j:TextType"/>
<xsd:element name="SentenceIssuer.Organization" type="j:OrganizationType"/>
<xsd:element name="StatusText" type="j:TextType"/>
<xsd:element name="StatuteCodeID" type="j:IDType"/>
<xsd:element name="StatuteOffenseID" type="j:IDType"/>
<xsd:element name="StatuteText" type="j:TextType"/>
<xsd:element name="StatuteJurisdiction" type="j:JurisdictionType"/>
<xsd:element name="StreetFullText" type="j:TextType"/>
<xsd:element name="StreetTypeText" type="j:TextType"/>
<xsd:element name="Subject" type="j:SubjectType"/>
<xsd:element name="SubjectCautionInformationCaveat" type="j:TextType"/>
<xsd:element name="SubjectID" type="j:IDType"/>
```



```
<xsd:element name="SubjectOffenderNoticeCaveat" type="j:TextType"/>
<xsd:element name="Supervision" type="j:SupervisionType"/>
<xsd:element name="SupervisionAgency" type="j:OrganizationType"/>
<xsd:element name="SupervisionCustodyStatus" type="j:StatusType"/>
<xsd:element name="SupervisionSubject" type="j:SubjectType"/>
<xsd:element name="SupervisionRelease" type="j:ReleaseType"/>
<xsd:element name="TelephoneNumberFullID" type="j-xsd:string"/>
</xsd:schema>
```

## 5.0 Example Rap Sheet

The following example shows a rap sheet encoded in the XML format.

```
<?xml version="1.0"?>
<rap:RapSheet j:reportedDate="2004-11-08" xmlns:rap="http://www.nlets.org/rapsheet/3.0.0"
xmlns:j="http://www.it.ojp.gov/jxdm/3.0" version="3.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.nlets.org/rapsheet/3.0.0 file:///C:/Rapsheet/DocExt.xsd">
  <rap:Introduction>
    <rap:RapSheetRequest id="0HJGTM1THN">
      <j:PersonFBIID>
        <j:ID>900660</j:ID>
      </j:PersonFBIID>
      <j:PersonStateID>
        <j:ID>CA99999978</j:ID>
        <j:IDIssuingAuthorityText>CA</j:IDIssuingAuthorityText>
      </j:PersonStateID>
      <rap:PurposeCode>C</rap:PurposeCode>
      <rap:Attention>AL 911137</rap:Attention>
    </rap:RapSheetRequest>
    <rap:Caveat>
      <j:CaveatText>This record is provided in response to your request. Use of the information contained in this
record is governed by state and federal law.</j:CaveatText>
      <j:CaveatReferenceDate>2002-02-08</j:CaveatReferenceDate>
      <j:CaveatIssuingAuthorityName>CA</j:CaveatIssuingAuthorityName>
    </rap:Caveat>
    <rap:ControlData rap:typeText="Request Index Number">273850293858</rap:ControlData>
    <rap:ControlData rap:typeText="Mail Rapsheet To">
```

Crime Information Bureau  
PO Box 2718

Madison, WI 53701-2718

```
</rap:ControlData>
  </rap:Introduction>
  <j:Subject>
    <j:PersonName>
      <j:PersonPrefixName>Mr</j:PersonPrefixName>
      <j:PersonGivenName>John</j:PersonGivenName>
      <j:PersonMiddleName>P</j:PersonMiddleName>
      <j:PersonSurName>Jones</j:PersonSurName>
      <j:PersonSuffixName>Jr</j:PersonSuffixName>
    </j:PersonName>
    <j:PersonAlternateName>
      <j:PersonGivenName>Bill</j:PersonGivenName>
      <j:PersonSurName>Williams</j:PersonSurName>
    </j:PersonAlternateName>
    <j:Residence j:reportedDate="1987-09-15">
      <j:LocationAddress>
        <j:AddressFullText>1565 N Park Place, Hamilton, NJ 08610</j:AddressFullText>
      </j:LocationAddress>
    </j:Residence>
    <j:Residence j:reportedDate="1989-04-22">
      <j:LocationAddress>
        <j:LocationStreet>
          <j:StreetFullText>1565 N Park Place</j:StreetFullText>
        </j:LocationStreet>
        <j:LocationSecondaryUnitText>Apt B</j:LocationSecondaryUnitText>
        <j:LocationCityName>Hamilton</j:LocationCityName>
        <j:LocationStateName>NJ</j:LocationStateName>
        <j:LocationPostalCodeID>
          <j:ID>08610-1234</j:ID>
        </j:LocationPostalCodeID>
      </j:LocationAddress>
    </j:Residence>
  </j:Subject>
</rap:ControlData>
```

```
</j:LocationAddress>
<j:LocationContactInformation>
  <j>ContactTelephoneNumber>
    <j:TelephoneNumberFullID>608-266-0872</j:TelephoneNumberFullID>
  </j>ContactTelephoneNumber>
</j:LocationContactInformation>
</j:Residence>
<j:Employment j:reportedDate="1987-09-15">
  <j:EmploymentEmployerName>Carful Cabs</j:EmploymentEmployerName>
  <j:EmploymentOccupationText>Taxi Driver</j:EmploymentOccupationText>
  <j:EmploymentLocation>
    <j:LocationAddress>
      <j:AddressFullText>1565 N Park Place, Hamilton, NJ 08610</j:AddressFullText>
    </j:LocationAddress>
    <j:LocationContactInformation>
      <j>ContactTelephoneNumber>
        <j:TelephoneNumberFullID>608-242-2000</j:TelephoneNumberFullID>
      </j>ContactTelephoneNumber>
    </j:LocationContactInformation>
  </j:EmploymentLocation>
</j:Employment>
<j:PersonBirthDate>1953-02-12</j:PersonBirthDate>
<j:PersonBirthLocation>
  <j:LocationName>Virginia</j:LocationName>
</j:PersonBirthLocation>
<j:PersonBirthLocation>
  <j:LocationName>Unknown</j:LocationName>
</j:PersonBirthLocation>
<j:PersonDeathDate>2002-05-12</j:PersonDeathDate>
<j:PersonLivingIndicator j:reportingOrganizationText="WI041015Y" j:reportedDate="2001-02-27"
j:commentText="Fingerprint Supported">>false</j:PersonLivingIndicator>
```

```
<j:PersonRegisteredOffenderIndicator
j:reportingOrganizationText="Arkansas">true</j:PersonRegisteredOffenderIndicator>
  <j:PersonFirearmSalesDisqualifiedIndicator j:reportingOrganizationText="Minnesota"
j:commentText="Disqualified">true</j:PersonFirearmSalesDisqualifiedIndicator>
  <j:PersonAssignedIDDetails>
    <j:PersonSSNID>
      <j:ID>220565860</j:ID>
    </j:PersonSSNID>
    <j:PersonDriverLicenseID>
      <j:ID>C4556289248R</j:ID>
      <j:IDIssuingAuthorityText>WI</j:IDIssuingAuthorityText>
    </j:PersonDriverLicenseID>
    <j:PersonFBIID>
      <j:ID>62660CA12</j:ID>
    </j:PersonFBIID>
    <j:PersonStateID>
      <j:ID>CA9936278</j:ID>
      <j:IDIssuingAuthorityText>CA</j:IDIssuingAuthorityText>
    </j:PersonStateID>
    <j:PersonOtherID>
      <j:ID>742786</j:ID>
      <j:IDTypeText>Correctional ID</j:IDTypeText>
      <j:IDIssuingAuthorityText>IL</j:IDIssuingAuthorityText>
    </j:PersonOtherID>
    <j:PersonOtherID>
      <j:ID>C4556289248R</j:ID>
      <j:IDTypeText>State ID Card</j:IDTypeText>
      <j:IDIssuingAuthorityText>WI</j:IDIssuingAuthorityText>
    </j:PersonOtherID>
  </j:PersonAssignedIDDetails>
</j:PersonPhysicalDetails>
```

```

    <j:PersonHeightMeasure j:personHeightUnitCode="ncic" j:reportedDate="1994-07-04">601</j:PersonHeightMeasure>
    <j:PersonHeightMeasure j:personHeightUnitCode="ncic" j:reportedDate="1994-07-04">602</j:PersonHeightMeasure>
    <j:PersonWeightMeasure j:personWeightUnitCode="ncic" j:reportedDate="1997-03-21">180</j:PersonWeightMeasure>
    <j:PersonWeightMeasure j:personWeightUnitCode="ncic" j:reportedDate="1997-03-21">190</j:PersonWeightMeasure>
    <j:PersonEyeColorText j:reportedDate="1988-03-21">Blue</j:PersonEyeColorText>
    <j:PersonEyeColorText j:reportedDate="1988-03-21">Brown</j:PersonEyeColorText>
    <j:PersonHairColorText j:reportedDate="1989-07-04">Brown</j:PersonHairColorText>
    <j:PersonHairColorText j:reportedDate="1989-07-04">Black</j:PersonHairColorText>
    <j:PersonSexText>Female</j:PersonSexText>
    <j:PersonRaceText>White</j:PersonRaceText>
    <j:PersonRaceText>Black</j:PersonRaceText>
    <j:PersonSkinToneText>Albino</j:PersonSkinToneText>
    <j:PersonSkinToneText>Dark</j:PersonSkinToneText>
    <j:PersonPhysicalFeature j:reportedDate="1999-06-04">
      <j:PhysicalFeatureCategoryText>tattoo</j:PhysicalFeatureCategoryText>
      <j:PhysicalFeatureTypeCode>TAT FARM</j:PhysicalFeatureTypeCode>
      <j:PhysicalFeatureDescriptionText>Dragon tattoo on right forearm.</j:PhysicalFeatureDescriptionText>
      <j:PhysicalFeatureImage>
        <j:BinaryObject.Base64>dkriJEORjSo9rj6+/kDIslDK</j:BinaryObject.Base64>
        <j:BinaryFormatText>image/jpeg</j:BinaryFormatText>
        <j:BinarySizeValue>55</j:BinarySizeValue>
        <j:BinaryTypeText>Tattoo</j:BinaryTypeText>
        <j:BinaryDescriptionText>Color pic</j:BinaryDescriptionText>
        <j:BinaryReferenceID>
          <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
        </j:BinaryReferenceID>
        <j:BinaryCaptureDate>1999-06-04</j:BinaryCaptureDate>
      </j:PhysicalFeatureImage>
    </j:PersonPhysicalFeature>
  </j:Person>

```

```
<j:BinaryCaptureOrganization>
  <j:OrganizationName>CIB</j:OrganizationName>
  <j:OrganizationORIID>
    <j:ID>WI013415Y</j:ID>
  </j:OrganizationORIID>
</j:BinaryCaptureOrganization>
</j:PhysicalFeatureImage>
</j:PersonPhysicalFeature>
</j:PersonPhysicalDetails>
<j:PersonSocialDetails>
  <j:PersonEthnicityText>Hispanic Or Latino</j:PersonEthnicityText>
  <j:PersonCitizenshipText j:reportedDate="2001-11-29">United States</j:PersonCitizenshipText>
  <j:PersonMaritalStatusText j:reportedDate="2001-11-29">Never Married</j:PersonMaritalStatusText>
  <j:PersonReligionText j:reportedDate="2001-11-29">Lutheran</j:PersonReligionText>
</j:PersonSocialDetails>
<j:PersonBiometricDetails>
  <j:PersonDigitalImage>
    <j:BinaryObject.Base64>dkriJEORjSo9rj6+/kDIslDK</j:BinaryObject.Base64>
    <j:BinaryFormatText>image/jpeg</j:BinaryFormatText>
    <j:BinarySizeValue>15</j:BinarySizeValue>
    <j:BinaryTypeText>Identification</j:BinaryTypeText>
    <j:BinaryDescriptionText>Color pic</j:BinaryDescriptionText>
    <j:BinaryReferenceID>
      <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
    </j:BinaryReferenceID>
    <j:BinaryCaptureDate>1999-06-04</j:BinaryCaptureDate>
    <j:BinaryCaptureOrganization>
      <j:OrganizationName>CIB</j:OrganizationName>
      <j:OrganizationORIID>
        <j:ID>WI013415Y</j:ID>
      </j:OrganizationORIID>
    </j:BinaryCaptureOrganization>
  </j:PersonDigitalImage>
</j:PersonBiometricDetails>
</j:PersonSocialDetails>
</j:PersonPhysicalDetails>
</j:PersonSocialDetails>
</j:PersonPhysicalFeature>
</j:PhysicalFeatureImage>
</j:BinaryCaptureOrganization>
```

```

    </j:BinaryCaptureOrganization>
  </j:PersonDigitalImage>
</j:PersonDigitalImage>
  <j:BinaryTypeText>Mugshot</j:BinaryTypeText>
  <j:BinaryCaptureOrganization>
    <j:OrganizationName>Wisconsin</j:OrganizationName>
    <j:OrganizationORIID>
      <j:ID>WI013415Y</j:ID>
    </j:OrganizationORIID>
  </j:BinaryCaptureOrganization>
</j:PersonDigitalImage>
<j:PersonBloodTypeText>O Positive</j:PersonBloodTypeText>
<j:PersonFingerprintSet>
  <j:Fingerprint>
    <j:BiometricImage>
      <j:BinaryObject.Base64>dkriJEORjSo9rj6+/kDIslDK</j:BinaryObject.Base64>
      <j:BinaryFormatText>image/wsq</j:BinaryFormatText>
      <j:BinarySizeValue>75</j:BinarySizeValue>
      <j:BinaryTypeText>Fingerprint</j:BinaryTypeText>
      <j:BinaryDescriptionText>Grayscale</j:BinaryDescriptionText>
      <j:BinaryReferenceID>
        <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
      </j:BinaryReferenceID>
      <j:BinaryCaptureDate>1999-06-04</j:BinaryCaptureDate>
      <j:BinaryCaptureOrganization>
        <j:OrganizationName>CIB</j:OrganizationName>
        <j:OrganizationORIID>
          <j:ID>WI013415Y</j:ID>
        </j:OrganizationORIID>
      </j:BinaryCaptureOrganization>
    </j:BiometricImage>
  </j:Fingerprint>
</j:PersonFingerprintSet>
</j:PersonBloodTypeText>
</j:PersonDigitalImage>
</j:PersonDigitalImage>

```



```

        <j:FingerprintFingerText>Right Index</j:FingerprintFingerText>
    </j:Fingerprint>
</j:PersonFingerprintSet>
<j:PersonFingerprintSet>
    <j:BiometricValueText>66AA09TTPI58AA6413XI</j:BiometricValueText>
    <j:BiometricEncodingMethodText>FPC</j:BiometricEncodingMethodText>
</j:PersonFingerprintSet>
<j:PersonDNA>
    <j:BiometricCaptureOrganization>
        <j:OrganizationName>Wisconsin</j:OrganizationName>
        <j:OrganizationORIID>
            <j:ID>WI041015Y</j:ID>
        </j:OrganizationORIID>
    </j:BiometricCaptureOrganization>
</j:PersonDNA>
<j:PersonDNA>
    <j:BiometricEncodingMethodText>STR</j:BiometricEncodingMethodText>
    <j:BiometricDescriptionText>Pursuant to sex offender registration.</j:BiometricDescriptionText>
    <j:BiometricCaptureOrganization>
        <j:OrganizationName>Crime Laboratory</j:OrganizationName>
        <j:OrganizationORIID>
            <j:ID>WI041015Y</j:ID>
        </j:OrganizationORIID>
    </j:BiometricCaptureOrganization>
    <j:BiometricCaptureDate>2001-02-27</j:BiometricCaptureDate>
    <j:DNALocus>
        <j:DNALocusTypeText>Amelogenin</j:DNALocusTypeText>
        <j:DNALocusValue>X</j:DNALocusValue>
        <j:DNALocusValue>Y</j:DNALocusValue>
    </j:DNALocus>
</j:DNALocus>

```

```
<j:DNALocusTypeText>CSF1PO</j:DNALocusTypeText>
<j:DNALocusValue>11</j:DNALocusValue>
<j:DNALocusValue>12</j:DNALocusValue>
</j:DNALocus>
<j:DNALocus>
  <j:DNALocusTypeText>D13S317</j:DNALocusTypeText>
  <j:DNALocusValue>11</j:DNALocusValue>
</j:DNALocus>
<j:DNALocus>
  <j:DNALocusTypeText>vWA</j:DNALocusTypeText>
  <j:DNALocusValue>14</j:DNALocusValue>
  <j:DNALocusValue>16</j:DNALocusValue>
</j:DNALocus>
</j:PersonDNA>
<j:PersonPalmPrint>
  <j:BiometricImage>
    <j:BinaryObject.Base64>dkriJEORjSo9rj6+/kDIslDK</j:BinaryObject.Base64>
    <j:BinaryFormatText>image/wsqr</j:BinaryFormatText>
    <j:BinarySizeValue>75</j:BinarySizeValue>
    <j:BinaryTypeText>Right Palm</j:BinaryTypeText>
    <j:BinaryDescriptionText>Grayscale</j:BinaryDescriptionText>
    <j:BinaryReferenceID>
      <j:ID>http://www.doj.state.wi.us/les/XML/files/plym0070.jpg</j:ID>
    </j:BinaryReferenceID>
    <j:BinaryCaptureDate>1999-06-04</j:BinaryCaptureDate>
    <j:BinaryCaptureOrganization>
      <j:OrganizationName>CIB</j:OrganizationName>
      <j:OrganizationORIID>
        <j:ID>WI013415Y</j:ID>
      </j:OrganizationORIID>
    </j:BinaryCaptureOrganization>
```

```

        </j:BiometricImage>
    </j:PersonPalmPrint>
    <j:PersonPalmPrint>
        <j:BiometricImage>
            <j:BinaryTypeText>Both Palms</j:BinaryTypeText>
            <j:BinaryCaptureDate>1999-06-04</j:BinaryCaptureDate>
            <j:BinaryCaptureOrganization>
                <j:OrganizationName>CIB</j:OrganizationName>
                <j:OrganizationORIID>
                    <j:ID>WI013415Y</j:ID>
                </j:OrganizationORIID>
            </j:BinaryCaptureOrganization>
        </j:BiometricImage>
    </j:PersonPalmPrint>
</j:PersonBiometricDetails>
<j:PersonMedicalDetails>
    <j:PersonMedicalCondition j:reportedDate="1964-04-01">
        <j:MedicalConditionText>Suicidal</j:MedicalConditionText>
    </j:PersonMedicalCondition>
    <j:PersonMedicalCondition j:reportedDate="1969-10-16">
        <j:MedicalConditionText>Diabetic</j:MedicalConditionText>
    </j:PersonMedicalCondition>
</j:PersonMedicalDetails>
    <j:SubjectCautionInformationCaveat j:reportingOrganizationText="CA" j:reportedDate="1995-11-02">Armed and
dangerous</j:SubjectCautionInformationCaveat>
    <j:SubjectOffenderNoticeCaveat j:reportingOrganizationText="CA" j:reportedDate="1995-11-02">Subject of the
record is a convicted felon.</j:SubjectOffenderNoticeCaveat>
</j:Subject>
<rap:Cycle>
    <rap:CycleEarliestDate>1995-07-04</rap:CycleEarliestDate>
    <rap:CycleTrackingID>

```

```
<j:ID>46019527</j:ID>
</rap:CycleTrackingID>
<j:Incident>
  <j:ActivityDate>1998-05-28</j:ActivityDate>
</j:Incident>
<j:Arrest>
  <j:ActivityTypeText>Adult</j:ActivityTypeText>
  <j:ActivityDate>1998-05-30</j:ActivityDate>
  <j:ActivityCommentText>Resisted arrest</j:ActivityCommentText>
  <j:ArrestAgencyRecordID>
    <j:ID>1998AF002354</j:ID>
  </j:ArrestAgencyRecordID>
  <j:ArrestSubject>
    <j:PersonName>
      <j:PersonFullName>Mitch Doherty</j:PersonFullName>
    </j:PersonName>
    <j:PersonName>
      <j:PersonFullName>Michael Doherty</j:PersonFullName>
    </j:PersonName>
    <j:SubjectID>
      <j:ID>48204395</j:ID>
    </j:SubjectID>
  </j:ArrestSubject>
  <j:ArrestCharge>
    <j:ChargeID>
      <j:ID>94D002356</j:ID>
    </j:ChargeID>
    <j:ChargeText>Robbery</j:ChargeText>
    <j:ChargeSequenceID>
      <j:ID>01</j:ID>
    </j:ChargeSequenceID>
```

```
<j:ChargeTrackingID>
  <j:ID>AD486</j:ID>
</j:ChargeTrackingID>
<j:ChargeDescriptionText>Referred for clinical evaluation</j:ChargeDescriptionText>
<j:ChargeClassification>
  <j:ChargeApplicabilityText>Attempt</j:ChargeApplicabilityText>
  <j:ChargeReducingFactorText>First offense</j:ChargeReducingFactorText>
  <j:ChargeSpecialAllegationText>Elderly victim</j:ChargeSpecialAllegationText>
  <j:ChargeSeverityText>Felony</j:ChargeSeverityText>
</j:ChargeClassification>
<j:ChargeCountQuantity>4</j:ChargeCountQuantity>
<j:ChargeDisposition j:commentText="Charges Dropped">
  <j:ChargeDispositionDate>1996-07-07</j:ChargeDispositionDate>
  <j:ChargeDispositionDescriptionText>Case closed without referral for
prosecution.</j:ChargeDispositionDescriptionText>
</j:ChargeDisposition>
<j:ChargeReporting>
  <j:ChargeNCICCode>0106</j:ChargeNCICCode>
</j:ChargeReporting>
<j:ChargeStatute>
  <j:StatuteCodeID>
    <j:ID>943.2(e)10</j:ID>
  </j:StatuteCodeID>
  <j:StatuteOffenseID>
    <j:ID>20836</j:ID>
  </j:StatuteOffenseID>
  <j:StatuteText>Statute Text</j:StatuteText>
  <j:StatuteJurisdiction>
    <j:LocationStateName>WI</j:LocationStateName>
  </j:StatuteJurisdiction>
</j:ChargeStatute>
```

```
</j:ArrestCharge>
<j:ArrestAgency>
  <j:OrganizationName>CIB</j:OrganizationName>
  <j:OrganizationORIID>
    <j:ID>WI013415Y</j:ID>
  </j:OrganizationORIID>
</j:ArrestAgency>
</j:Arrest>
<j:Booking>
  <j:BookingAgencyRecordID>
    <j:ID>1998AF002355</j:ID>
  </j:BookingAgencyRecordID>
  <j:BookingAgency>
    <j:OrganizationName>CIB</j:OrganizationName>
    <j:OrganizationORIID>
      <j:ID>WI013415Y</j:ID>
    </j:OrganizationORIID>
  </j:BookingAgency>
</j:Booking>
<rap:Prosecution>
  <j:ActivityDate>1998-05-30</j:ActivityDate>
  <rap:ProsecutionAgencyRecordID>
    <j:ID>1998AF002356</j:ID>
  </rap:ProsecutionAgencyRecordID>
  <rap:ProsecutionSubject>
    <j:PersonName>
      <j:PersonFullName>Mitch Doherty</j:PersonFullName>
    </j:PersonName>
    <j:PersonName>
      <j:PersonFullName>Michael Doherty</j:PersonFullName>
    </j:PersonName>
  </rap:ProsecutionSubject>
</rap:Prosecution>
```

```

    <j:SubjectID>
      <j:ID>48204396</j:ID>
    </j:SubjectID>
  </rap:ProsecutionSubject>
  <j:ProsecutionCharge>
    <j:ChargeID>
      <j:ID>94D002356</j:ID>
    </j:ChargeID>
    <j:ChargeText>Burglary</j:ChargeText>
    <j:ChargeSequenceID>
      <j:ID>01</j:ID>
    </j:ChargeSequenceID>
    <j:ChargeTrackingID>
      <j:ID>AD486</j:ID>
    </j:ChargeTrackingID>
    <j:ChargeDescriptionText>Referred for clinical evaluation</j:ChargeDescriptionText>
    <j:ChargeClassification>
      <j:ChargeApplicabilityText>Attempt</j:ChargeApplicabilityText>
      <j:ChargeReducingFactorText>First offense</j:ChargeReducingFactorText>
      <j:ChargeSpecialAllegationText>Elderly victim</j:ChargeSpecialAllegationText>
      <j:ChargeSeverityText>Felony</j:ChargeSeverityText>
    </j:ChargeClassification>
    <j:ChargeCountQuantity>4</j:ChargeCountQuantity>
    <j:ChargeDisposition j:commentText="Not Prosecuted">
      <j:ChargeDispositionDate>1996-07-07</j:ChargeDispositionDate>
      <j:ChargeDispositionDescriptionText>Case closed without court
filing.</j:ChargeDispositionDescriptionText>
    </j:ChargeDisposition>
    <j:ChargeReporting>
      <j:ChargeNCICCode>0106</j:ChargeNCICCode>
    </j:ChargeReporting>
  </j:ProsecutionCharge>

```

```
<j:ChargeStatute>
  <j:StatuteCodeID>
    <j:ID>943.2(e)10</j:ID>
  </j:StatuteCodeID>
  <j:StatuteOffenseID>
    <j:ID>20836</j:ID>
  </j:StatuteOffenseID>
  <j:StatuteText>Text</j:StatuteText>
  <j:StatuteJurisdiction>
    <j:LocationStateName>WI</j:LocationStateName>
  </j:StatuteJurisdiction>
</j:ChargeStatute>
</j:ProsecutionCharge>
<rap:ProsecutionAgency>
  <j:OrganizationName>CIB</j:OrganizationName>
  <j:OrganizationORIID>
    <j:ID>WI013415Y</j:ID>
  </j:OrganizationORIID>
</rap:ProsecutionAgency>
</rap:Prosecution>
<rap:CourtAction>
  <j:ActivityDate>1998-05-30</j:ActivityDate>
  <j:ActivityCommentText>Jury trial waived.</j:ActivityCommentText>
  <rap:CourtRecordID>
    <j:ID>1998AF002356</j:ID>
  </rap:CourtRecordID>
  <rap:CourtSubject>
    <j:PersonName>
      <j:PersonFullName>Mitch Doherty</j:PersonFullName>
    </j:PersonName>
    <j:PersonName>
      <j:PersonFullName>
    </j:PersonName>
  </rap:CourtSubject>
</rap:CourtAction>
</rap:CourtAction>
```



```
        <j:PersonFullName>Michael Doherty</j:PersonFullName>
    </j:PersonName>
    <j:SubjectID>
        <j:ID>48204397</j:ID>
    </j:SubjectID>
</rap:CourtSubject>
<j:CourtCharge>
    <j:ChargeID>
        <j:ID>94D002356</j:ID>
    </j:ChargeID>
    <j:ChargeText>Trespassing</j:ChargeText>
    <j:ChargeSequenceID>
        <j:ID>01</j:ID>
    </j:ChargeSequenceID>
    <j:ChargeTrackingID>
        <j:ID>AD486</j:ID>
    </j:ChargeTrackingID>
    <j:ChargeDescriptionText>Referred for clinical evaluation</j:ChargeDescriptionText>
    <j:ChargeClassification>
        <j:ChargeApplicabilityText>Attempt</j:ChargeApplicabilityText>
        <j:ChargeReducingFactorText>First offense</j:ChargeReducingFactorText>
        <j:ChargeSpecialAllegationText>Elderly victim</j:ChargeSpecialAllegationText>
        <j:ChargeSeverityText>Felony</j:ChargeSeverityText>
    </j:ChargeClassification>
    <j:ChargeCountQuantity>4</j:ChargeCountQuantity>
    <j:ChargeDisposition j:commentText="Convicted">
        <j:ChargeDispositionDate>1996-07-07</j:ChargeDispositionDate>
        <j:ChargeDispositionDescriptionText>Guilty.</j:ChargeDispositionDescriptionText>
    </j:ChargeDisposition>
    <j:ChargeReporting>
        <j:ChargeNCICCode>0106</j:ChargeNCICCode>
```

```
</j:ChargeReporting>
<j:ChargeStatute>
  <j:StatuteCodeID>
    <j:ID>943.2(e)10</j:ID>
  </j:StatuteCodeID>
  <j:StatuteOffenseID>
    <j:ID>20836</j:ID>
  </j:StatuteOffenseID>
  <j:StatuteText>Text</j:StatuteText>
  <j:StatuteJurisdiction>
    <j:LocationStateName>WI</j:LocationStateName>
  </j:StatuteJurisdiction>
</j:ChargeStatute>
</j:CourtCharge>
<j:Court>
  <j:OrganizationName>Portage County Circuit Court</j:OrganizationName>
  <j:OrganizationORIID>
    <j:ID>WI014400Y</j:ID>
  </j:OrganizationORIID>
</j:Court>
</rap:CourtAction>
<rap:Sentencing>
  <j:ActivityDate>1998-05-30</j:ActivityDate>
  <j:ActivityCommentText>Psychiatric evaluation ordered</j:ActivityCommentText>
  <rap:CourtRecordID>
    <j:ID>1998AF002356</j:ID>
  </rap:CourtRecordID>
  <j:Sentence>
    <j:SentenceCharge>
      <j:ChargeID>
        <j:ID>94D002356</j:ID>
      </j:ChargeID>
    </j:SentenceCharge>
  </j:Sentence>
</rap:Sentencing>
</j:ChargeReporting>
```

```
        <j:ChargeID>
        <j:ChargeText>Trespassing</j:ChargeText>
        <j:ChargeSequenceID>
            <j:ID>01</j:ID>
        </j:ChargeSequenceID>
        <j:ChargeTrackingID>
            <j:ID>AD486</j:ID>
        </j:ChargeTrackingID>
    </j:SentenceCharge>
    <j:SentenceDescriptionText>3Y Prison</j:SentenceDescriptionText>
    <j:SentenceDescriptionText>$500 fine and costs</j:SentenceDescriptionText>
</j:Sentence>
<j:Court>
    <j:OrganizationName>Portage County Circuit Court</j:OrganizationName>
    <j:OrganizationORIID>
        <j:ID>WI014400Y</j:ID>
    </j:OrganizationORIID>
</j:Court>
</rap:Sentencing>
<rap:Supervision>
    <j:ActivityDate>1998-05-30</j:ActivityDate>
    <j:ActivityCommentText>Probation revoked. </j:ActivityCommentText>
    <rap:SupervisionAgencyRecordID>
        <j:ID>1998AF002356</j:ID>
    </rap:SupervisionAgencyRecordID>
    <rap:CourtRecordID>
        <j:ID>1998AF002356</j:ID>
    </rap:CourtRecordID>
    <j:SupervisionSubject>
        <j:PersonName>
            <j:PersonFullName>Mitch Doherty</j:PersonFullName>
        </j:PersonName>
    </j:SupervisionSubject>
</rap:Supervision>
</rap:Person>
</rap:Case>
</rap:Case>
```

```

    </j:PersonName>
    <j:PersonName>
      <j:PersonFullName>Michael Doherty</j:PersonFullName>
    </j:PersonName>
    <j:SubjectID>
      <j:ID>48204395</j:ID>
    </j:SubjectID>
  </j:SupervisionSubject>
  <j:SupervisionAgency>
    <j:OrganizationName>Portage County Jail</j:OrganizationName>
    <j:OrganizationORIID>
      <j:ID>WI014400Y</j:ID>
    </j:OrganizationORIID>
  </j:SupervisionAgency>
  <j:SupervisionCustodyStatus>
    <j:StatusText>incarcerated</j:StatusText>
  </j:SupervisionCustodyStatus>
  <j:SupervisionRelease>
    <j:ActivityDate>1998-06-30</j:ActivityDate>
  </j:SupervisionRelease>
</rap:Supervision>
</rap:Cycle>
<j:Organization>
  <j:OrganizationName>Lodi Police Department</j:OrganizationName>
  <j:OrganizationORIID>
    <j:ID>WI0111000</j:ID>
  </j:OrganizationORIID>
  <j:OrganizationLocation>
    <j:LocationAddress>
      <j:AddressFullText>1565 N Park Place, Hamilton, NJ 08610
</j:AddressFullText>
```

```
<j:LocationStreet>
  <j:StreetFullText>1565 N Park Place</j:StreetFullText>
</j:LocationStreet>
<j:LocationCityName>Hamilton</j:LocationCityName>
<j:LocationStateName>NJ</j:LocationStateName>
<j:LocationPostalCodeID>
  <j:ID>08610</j:ID>
</j:LocationPostalCodeID>
</j:LocationAddress>
<j:LocationContactInformation>
  <j:ContactInformationDescriptionText>George</j:ContactInformationDescriptionText>
  <j:ContactTelephoneNumber>
    <j:TelephoneNumberFullID>4859304869</j:TelephoneNumberFullID>
  </j:ContactTelephoneNumber>
  <j:ContactFacsimileNumber>
    <j:TelephoneNumberFullID>4859304870</j:TelephoneNumberFullID>
  </j:ContactFacsimileNumber>
  <j:ContactEmailID>
    <j:ID>george@lodiPD.gov</j:ID>
  </j:ContactEmailID>
</j:LocationContactInformation>
</j:OrganizationLocation>
<j:OrganizationAbbreviationText>LOPD</j:OrganizationAbbreviationText>
<j:OrganizationUnitName>Records</j:OrganizationUnitName>
</j:Organization>
</rap:RapSheet>
```

# *Appendix A: XML Transformation*

## **5.1 Global Justice XML Data Model**

The Office of Justice Programs (OJP), together with the Global Justice Information Sharing Initiative (Global), has now released the [Global Justice Extensible Markup Language \(XML\) Data Model \(GJXDM\), Version 3.0](#), to the justice community. Developed by Global and OJP, the GJXDM is an object-oriented data model comprised of a well-defined vocabulary of approximately 2,500 stable data objects, or reusable components, that facilitate the exchange and reuse of information from multiple sources and multiple applications.

The GJXDM represents a significant milestone in the process of developing appropriate standards for expressing the baseline data needs of the justice and public safety communities and their related partners. In that sense, it will become a reference benchmark for jurisdictions in the justice and public safety communities to use, in its entirety or in part, to fulfill their specific needs.

The purpose of the GJXDM is to provide a consistent, extensible, maintainable XML schema reference specification for data elements and types that represent the data requirements of the general justice and public safety communities. A secondary goal is to provide a baseline model for the data dictionary that can be represented in advanced technologies beyond XML Schema.

Generally, electronic justice data exchange is accomplished via documents, queries, responses and other messages. In XML Schema terminology, all of these are considered XML documents. In the GJXDM, XML documents refer to standard business exchange containers (i.e. Rap Sheet, Arrest Warrant, or Incident Report) that are usually persistent. This means that they are archived, maintained, or reused, intact, by either the sender or receiver, XML documents can also be transactional. For example, queries, responses, or messages can be structured as XML documents. A document can be defined as ". . . something which brings together strongly related objects for a well defined business purpose or context . . ." or ". . . that bundle of data that is passed from one agency to another as part of an exchange . . ." [Gerry Coleman, Wisconsin Crime Information Bureau]. So, the term XML document can refer to a message or other form of information, as well as what we traditionally recognize as a document.

The GJXDM contains a DocumentType component from which standard XML document schemas can be derived. The GJXDM DocumentType includes commonly used properties such as control and records management metadata, security and classification metadata, and general document descriptive metadata.

Within its own target namespace, each justice reference document schema will import the GJXDM namespace (or a subset of it) and extend DocumentType for its root element. For example, a CitationDocument schema would import the GJXDM namespace and create (1) a CitationDocumentType that extends jxdm:DocumentType and (2) a complex root element CitationDocument of type CitationDocumentType. Consequently, CitationDocumentType inherits all of the standard metadata properties of jxdm:DocumentType, however, each document designates its own target namespace, enabling local document customization and preventing name conflicts.

Extensible Markup Language, or "XML," is a computer programming language designed to transmit both data and the meaning of the data. XML accomplishes this by being a markup language, a mechanism that identifies different structures within a document. Structured information contains both content (such as words, pictures, or video) and an indication of what role content plays, or its meaning. XML identifies

different structures by assigning data "tags" to define both the name of a data element and the format of the data within that element. Elements are combined to form objects.

An XML specification defines a standard way to add markup language to documents, identifying the embedded structures in a consistent way. By applying a consistent identification structure, data can be shared between different systems, up and down the levels of agencies, across the nation, and around the world, with the ease of using the Internet. In other words, XML lays the technological foundation that supports interoperability.

XML also allows structured relationships to be defined. The ability to represent objects and their relationships is key to creating a fully beneficial justice information sharing tool. A simple example can be used to illustrate this point:

A "person" object may contain elements like physical descriptors (e.g., eye and hair color, height, weight), biometric data (e.g., DNA, fingerprints), and social descriptors (e.g., marital status, occupation). A "vehicle" object would also contain many elements (such as description, registration, and/or lien-holder). The relationship between these two objects—person and vehicle—presents an interesting challenge that XML can address. Is the person the owner of the vehicle? The driver? Did he/she steal it? Get hit by it? And so forth.

XML is sanctioned by the World Wide Web Consortium (W3C), a premier forum comprised of agencies from across the globe committed to helping the World Wide Web reach its full potential by developing common protocols promoting Web evolution and interoperability. For more information on W3C, please visit <http://www.w3.org/>.

XML is compatible with major Internet transmission protocols, and is also highly compressible for faster transmission. Almost all major software vendors fully support the general XML standard. Major database vendors and their database applications provide software development "tools" to assist justice agency technical staff to develop and use XML more efficiently and productively within agency applications. XML is very developer-friendly, yet ordinary users with no particular XML expertise can make sense of an XML file. The XML standard is designed to be independent of vendor, operating system, source application, destination application, storage medium (database), and/or transport protocol.

This last fact makes XML great news for justice administrators: sharing vital information no longer entails purchasing new systems or compromising one's business practices. XML is generally recognized as an enabler for increasing the sharing of information, and has emerged as a key technology for assisting commercial and government organizations in exchanging information and conducting business over the Internet and intranets. XML is the "glue" that promotes interoperability—it allows systems already in use and those being developed to communicate with each other and paves the way for future expanded collaboration between agencies.

The GJXDM consists primarily of object classes and properties:

- Object classes are converted into XML Schema types. They represent a specific syntax and structure for a set of values.
- Properties are converted into XML Schema attributes and elements. They represent characteristics, or values of things.

The GJXDM v3 is a reference model that is based on a class hierarchy of many specific objects (xsd:type) derived from one very general object (SuperType). Each object contains any number of properties (xsd:element or xsd:attribute). These properties may be simple or complex elements (containing sub-elements). The highest level object classes (under the SuperType) are PersonType, OrganizationType,

PropertyType (i.e., things), LocationType, ContactInformationType (i.e., electronic means of contact), ActivityType, EventType, DocumentType, and other smaller supporting types. There is also a special class of RelationshipType used to specify meaning to a link between two object instances. On many of these objects, there are metadata properties (xsd:element or xsd:attribute) that supplement meaning. For example, all properties of MeasureType have a mandatory "units" attribute.

## **5.2 Sources**

**5.3** [http://it.ojp.gov/topic.jsp?topic\\_id=109](http://it.ojp.gov/topic.jsp?topic_id=109)

**5.4** <http://it.ojp.gov/documents/whatisXML.doc>

**5.5** <http://it.ojp.gov/jxdm/faq.html#N10079>

**5.6** <http://it.ojp.gov/jxdm/refnotes.html#N10079>



## ***Appendix B: XML Applications***

The XML criminal history can be used in two basic ways:

1. First, a standard scripting language can be used to transform the XML into another form or to extract essential elements from it. The prime example of scripting is using an XSL style sheet.
2. The second approach to using an XML document is through a program written in C, Java, or ECMAScript/Javascript. In this case, information is extracted from the XML document and used by the program. The primary means of accomplishing the data extraction is via a standardized approach such as one of the application programming interfaces (APIs) for the Document Object Model (DOM) or the Simple API for XML (SAX) although there are many others.

### **Scripting**

XSL transforms the tree structure of XML documents into another tree structure, typically a tree of HTML tags surrounding the same data content as in the XML file. The final step (although not strictly required) is to linearize the result tree into a sequential format in an output file. In the case of HTML output, this file is displayed in a web browser.

### **Direct Use via XML APIs**

XML parser software has been freely circulated on the internet for a number of years. There are several efforts underway to standardize on one or more APIs.

### **Simple API for XML (SAX) and Java API for XML Parsing (JAXP)**

The best known API is SAX and a related API from Sun Microsystems called JAXP (Java API for XML Parsing). This is a Java language-based API that uses a technique that calls a Java method (like a function or subprogram) as each XML element is parsed. The called method can then perform any action or build up any data structures that are needed. The SAX approach is simple to understand and program. It is similar to the approach used for graphical user interfaces. Since the SAX approach of invoking methods is under the control of the parser, it is more difficult to control the processing and a two-pass algorithm is often necessary. The first pass parses and records information from the XML document. The second pass then performs whatever action is needed or generates the result document.

## **Document Object Module (DOM)**

The DOM approach involves parsing the XML document into an internal data structure. The application program then can query the structure as needed to extract the desired information. The DOM approach is essentially following the two-pass algorithm mentioned in the preceding paragraph. DOM differs from SAX and JAXP in that DOM only specifies what information is made available and generally how the information is organized. There are DOM-compliant APIs for Java, C, JavaScript or ECMAScript that are not part of DOM itself.

Other parsers such as *Ælfred*, *XP*, and *MSXML* have been contributed by software developers. There are obvious advantages in using a published API or a widely used parser.

## **Validating and Non-validating Parsers**

XML parsers can be either validating or non-validating. Most parsers in use in applications are non-validating. Non-validating parsers can handle any XML that is well formed, that is, has start and end tags that are legally nested and evenly matched. For example, the document below on the left side is properly nested and the one on the right is not. Nesting is just one basic test of validity for an XML document.

A validating parser would make additional checks that certain elements appear and that only defined elements are present. The XML Schema definition provided in Section 4 is intended as a possible basis for validating an XML document. There are rules for criminal histories, however, that cannot be expressed in an XML Schema. For example, it is not possible to validate that a birthdate is more than 13 years in the past.

## Appendix C: Example Style Sheet

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- IMPORTANT: Do not manipulate the format of this document!
    The output format of the XSLT file is a positionally delimited,
    flat file that requires whitespace codes for proper transformations. -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:rap="http://www.nlets.org/rapsheet/3.0.0"
xmlns:j="http://www.it.ojp.gov/jxdm/3.0" >
    <xsl:template match="/">
<!--
        <xsl:value-of select="//rap:RapSheet/@Type"/>.<xsl:value-of select="//InitiatingORI"/>.<xsl:choose>
            <xsl:when test="//NLETSResponseHeader/receivedTime != "">
                <xsl:text/>
                <xsl:value-of select="//receivedTime"/>
                <xsl:text/>
                <xsl:value-of select="//receivedDate"/>
                <xsl:text/>
                <xsl:value-of select="//receivedLineMsgCount"/>
                <xsl:text/>
                <xsl:value-of select="//sendTime"/>
                <xsl:text/>
                <xsl:value-of select="//sendDate"/>
                <xsl:text/>
                <xsl:value-of select="//sendLineMsgCount"/>
                <xsl:text/>
                <xsl:for-each select="//NLETSResponseHeader/DestinationORI">
                    <xsl:choose>
                        <xsl:when test="position() = last()">
                            <xsl:value-of select="."/>
                        </xsl:when>
                        <xsl:otherwise>
                            <xsl:value-of select="."/>,</xsl:otherwise>
                    </xsl:choose>
                </xsl:for-each>
            </xsl:when>
            <xsl:otherwise>
                <xsl:text/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:template>
</!--
</xsl:stylesheet>
```

```

        </xsl:choose>
    </xsl:for-each>
    <xsl:text/>
    <xsl:if test="//ControlField != "">*<xsl:value-of select="//ControlField"/>
        <xsl:text/>
    </xsl:if>TXT</xsl:when>
<xsl:otherwise>
    <xsl:for-each select="//NLETSTResponseHeader/DestinationORI">
        <xsl:choose>
            <xsl:when test="position() = last()">
                <xsl:value-of select="."/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="."/>,</xsl:otherwise>
            </xsl:choose>
        </xsl:for-each>
    <xsl:if test="//ControlField != "">.*<xsl:value-of select="//ControlField"/>
    </xsl:if>.*TXT</xsl:otherwise>
</xsl:choose>
<xsl:text>#10;</xsl:text>
<xsl:if test="//NLETSTContent/Supplemental/FBINCICHeader">HDR/<xsl:value-of
select="//NLETSTContent/Supplemental/FBINCICHeader"/>
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="//rap:RapSheet/@Type = 'CR'">ATN/<xsl:value-of
select="//rap:RapSheet/Introduction/RapSheetRequest/Attention"/>
    <xsl:text>#10;</xsl:text>
</xsl:if> -->
<xsl:choose>
    <xsl:when test="//rap:RapSheet/Text">
        <xsl:value-of select="//rap:RapSheet/Text"/>

```

```

</xsl:when>
<xsl:otherwise>
  <!-- Global Variables -->
  <xsl:variable name="posCol1">0</xsl:variable>
  <!-- Position to start Column one -->
  <xsl:variable name="posCol2">24</xsl:variable>
  <!-- Position to start Column two -->
  <xsl:variable name="posCol3">48</xsl:variable>
  <!-- Position to start Column three -->
  <xsl:variable name="posMaxWidth">72</xsl:variable>
  <!-- Position for Maximum width -->
  <xsl:variable name="amtColSpacer">2</xsl:variable>
  <!-- Amount to space between Columns -->
  <xsl:text>***** CRIMINAL HISTORY RECORD
*****&#10;&#10;</xsl:text>
  <!-- dr: JTF 27-FEB-03: Date Printed (data as of date) added. -->
  <xsl:if test="//rap:RapSheet/@j:reportedDate">
    <xsl:text>Data As Of </xsl:text>
    <xsl:value-of select="//rap:RapSheet/@j:reportedDate"/>
    <xsl:text>&#10;&#10;</xsl:text>
  </xsl:if>
  <xsl:text>***** Introduction
*****&#10;&#10;</xsl:text>
  <xsl:text>This rap sheet was produced in response to the following
request:&#10;&#10;</xsl:text>
  <!-- ##### REQUESTER SECTION ##### -->
  <!-- Edited 05/03 -->
  <xsl:choose>
    <xsl:when
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonFullName) > 0">
      <xsl:text>Subject Name(s) </xsl:text>

```

```

<!-- dr: JTF 27-FEB-03: Added ability to do multiple names. -->
<xsl:for-each
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonFullName">
  <xsl:choose>
    <xsl:when test="position() = 1">
      <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text" select="."/>
        <xsl:with-param name="StartPos"
select="$posCol1"/>
        <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
        <xsl:with-param name="WrappedStartPos"
select="$posCol2"/>
        <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text" select="."/>
        <xsl:with-param name="StartPos"
select="$posCol2"/>
        <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
        <xsl:with-param name="WrappedStartPos"
select="$posCol2"/>
        <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>

```

```

select="$posMaxWidth + 1"/>
                                <xsl:with-param name="WrapWidth"
                                <xsl:with-param name="initialRun" select="0"/>
                                </xsl:call-template>
                                </xsl:otherwise>
                                <xsl:choose>
                                <xsl:text>#10;</xsl:text>
                                </xsl:for-each>

                                </xsl:when>
                                </xsl:choose>
                                <!-- -->
                                <!-- Edited 05/03 -->
                                <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonFBIID/j:ID) > 0">
                                <xsl:text>FBI Number </xsl:text>
                                <xsl:value-of
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonFBIID/j:ID"/>
                                <xsl:text>#10;</xsl:text>
                                </xsl:when>
                                </xsl:choose>
                                <!-- -->
                                <!-- Edited 05/03 -->
                                <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonStateID/j:ID) > 0">
                                <xsl:text>State Id Number </xsl:text>
                                <xsl:value-of
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonStateID/j:ID"/>
                                <xsl:text> (</xsl:text>

```

```

                                <xsl:value-of
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonStateID/j:IDIssuingAuthorityText"/>
                                <xsl:text>)&#10;</xsl:text>
                                </xsl:when>
                                </xsl:choose>
                                <!-- -->
                                <!-- Edited 05/03 -->
                                <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonSSNID/j:ID) &gt; 0">
                                <xsl:text>Social Security Number </xsl:text>
                                <xsl:for-each
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonSSNID/j:ID">
                                <xsl:choose>
                                <xsl:when test="position() = 1">
                                <xsl:call-template name="wrapIn">
                                <xsl:with-param name="Text" select="."/>
                                <xsl:with-param name="StartPos"
select="$posCol1"/>
                                <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
                                <xsl:with-param name="WrappedStartPos"
select="$posCol2"/>
                                <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
                                <xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>
                                <xsl:with-param name="initialRun" select="0"/>
                                </xsl:call-template>
                                </xsl:when>
                                <xsl:otherwise>

```



```

select="$posCol2"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
<xsl:call-template name="wrapIn">
  <xsl:with-param name="Text" select="."/>
  <xsl:with-param name="StartPos"
    <xsl:with-param name="EndPos"
    <xsl:with-param name="WrapedStartPos"
    <xsl:with-param name="CurrentPos"
    <xsl:with-param name="WrapWidth"
    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
</xsl:otherwise>
</xsl:choose>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
</xsl:when>
</xsl:choose>
<!-- -->
<!-- Edited 05/03 -->
<xsl:choose>
  <xsl:when
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonBirthDate) > 0">
  <xsl:text>Date of Birth </xsl:text>
  <xsl:for-each
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonBirthDate">
  <xsl:choose>
    <xsl:when test="position() = 1">
      <xsl:call-template name="wrapIn">

```

```

select="$posCol1"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

```

```

select="$posCol2"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

```

```

<xsl:with-param name="Text" select="."/>
<xsl:with-param name="StartPos"

<xsl:with-param name="EndPos"

<xsl:with-param name="WrapedStartPos"

<xsl:with-param name="CurrentPos"

<xsl:with-param name="WrapWidth"

<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text" select="."/>
    <xsl:with-param name="StartPos"

    <xsl:with-param name="EndPos"

    <xsl:with-param name="WrapedStartPos"

    <xsl:with-param name="CurrentPos"

    <xsl:with-param name="WrapWidth"

    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
</xsl:otherwise>
</xsl:choose>

```

```

                                <xsl:text>#10;</xsl:text>
                                </xsl:for-each>
                                <xsl:if
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonBirthDate)=0">
                                <xsl:text>#10;</xsl:text>
                                </xsl:if>
                                </xsl:when>
                                </xsl:choose>
                                <!-- -->
                                <!-- Edited 05/03 -->
                                <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/rap:ntroduction/rap:RapSheetRequest/j:PersonDriverLicenseID/j:ID) > 0">
                                <xsl:text>Driver's License Number </xsl:text>
                                <xsl:for-each
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonDriverLicenseID">
                                <xsl:choose>
                                <xsl:when test="position() = 1">
                                <xsl:call-template name="wrapIn">
                                <xsl:with-param name="Text" select="concat(j:ID,'
(j:IDIssuingAuthorityText,')'"/>
                                <xsl:with-param name="StartPos"
select="$posCol1"/>
                                <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
                                <xsl:with-param name="WrappedStartPos"
select="$posCol2"/>
                                <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
                                <xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>

```

```

        <xsl:with-param name="initialRun" select="0"/>
      </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text" select="concat(j:ID,'
        <xsl:with-param name="StartPos"
        <xsl:with-param name="EndPos"
        <xsl:with-param name="WrappedStartPos"
        <xsl:with-param name="CurrentPos"
        <xsl:with-param name="WrapWidth"
        <xsl:with-param name="initialRun" select="0"/>
      </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
  <xsl:text>#10;</xsl:text>
</xsl:for-each>
<xsl:if
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonDriverLicenseID)=0">
  <xsl:text>#10;</xsl:text>
</xsl:if>
</xsl:when>
</xsl:choose>
<!-- -->
<!-- Edited 05/03 -->

```

```

        <xsl:choose>
            <xsl:when
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonOtherID) &gt; 0">
                <xsl:text>Miscellaneous Number </xsl:text>
                <xsl:for-each
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonOtherID">
                    <xsl:choose>
                        <xsl:when test="position() = 1">
                            <xsl:call-template name="wrapIn">
                                <xsl:with-param name="Text" select="concat(j:ID,'
(,j:IDIssuingAuthorityText,','; ',j:IDTypeText,')')"/>
                                <xsl:with-param name="StartPos"
select="$posCol1"/>
                                <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
                                <xsl:with-param name="WrapedStartPos"
select="$posCol2"/>
                                <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
                                <xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>
                                <xsl:with-param name="initialRun" select="0"/>
                            </xsl:call-template>
                        </xsl:when>
                        <xsl:otherwise>
                            <xsl:call-template name="wrapIn">
                                <xsl:with-param name="Text" select="concat(j:ID,'
(,IDIssuingAuthority,','; ',j:IDTypeText,')')"/>
                                <xsl:with-param name="StartPos"
select="$posCol2"/>

```

```

select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
<xsl:if
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonOtherID)=0">
<xsl:text>#10;</xsl:text>
</xsl:if>
</xsl:when>
</xsl:choose>
<!-- -->
<!-- Edited 12/04 -->
<xsl:choose>
<xsl:when
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonSexText) > 0">
<xsl:text>Sex </xsl:text>
<xsl:for-each
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonSexText">
<xsl:choose>
<xsl:when test="position() = 1">
<xsl:call-template name="wrapIn">

```

```
select="$posCol1"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
<xsl:with-param name="Text" select="."/>
```

```
<xsl:with-param name="StartPos"
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun" select="0"/>
```

```
</xsl:call-template>
```

```
</xsl:when>
```

```
<xsl:otherwise>
```

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text" select="."/>
```

```
<xsl:with-param name="StartPos"
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun" select="0"/>
```

```
</xsl:call-template>
```

```
</xsl:otherwise>
```

```
</xsl:choose>
```





```

select="$posCol2"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text" select="."/>
    <xsl:with-param name="StartPos"

    <xsl:with-param name="EndPos"

    <xsl:with-param name="WrappedStartPos"

    <xsl:with-param name="CurrentPos"

    <xsl:with-param name="WrapWidth"

    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
</xsl:otherwise>
</xsl:choose>
  <xsl:text>#10;</xsl:text>
</xsl:for-each>
<xsl:if
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/j:PersonRaceText)=0">
  <xsl:text>#10;</xsl:text>
</xsl:if>
</xsl:when>
</xsl:choose>
<!-- -->
<!-- Edited 12/04 -->
<xsl:choose>

```

```

0">
    <xsl:when test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/@id) > 0">
        <xsl:text>Request Id          </xsl:text>
        <xsl:value-of
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/@id"/>
        <xsl:text>#10;</xsl:text>
    </xsl:when>
</xsl:choose>
<!-- -->
<!-- Edited 05/03 -->
<xsl:choose>
    <xsl:when
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/rap:PurposeCode) > 0">
        <xsl:text>Purpose Code      </xsl:text>
        <xsl:value-of
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/rap:PurposeCode"/>
        <xsl:text>#10;</xsl:text>
    </xsl:when>
</xsl:choose>
<!-- -->
<!-- Edited 05/03 -->
<xsl:choose>
    <xsl:when
test="count(//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/rap:Attention) > 0">
        <xsl:text>Attention          </xsl:text>
        <xsl:value-of
select="//rap:RapSheet/rap:Introduction/rap:RapSheetRequest/rap:Attention"/>
        <xsl:text>#10;</xsl:text>
    </xsl:when>
</xsl:choose>
<!-- -->

```

```

<xsl:text>#10;</xsl:text>
<xsl:text>The information in this rap sheet is subject to the following
caveats:#10;#10;</xsl:text>
<xsl:for-each select="//rap:RapSheet/rap:Introduction/rap:Caveat/j:CaveatText">
  <xsl:variable name="CaveatData">
    <xsl:value-of select="normalize-
space(//rap:RapSheet/rap:Introduction/rap:Caveat/j:CaveatText)"/>
    <xsl:text> (</xsl:text>
    <xsl:value-of
select="//rap:RapSheet/rap:Introduction/rap:Caveat/j:CaveatIssuingAuthorityName"/>
    <xsl:if
test="//rap:RapSheet/rap:Introduction/rap:Caveat/j:CaveatReferenceDate">
      <xsl:text>; </xsl:text>
      <xsl:value-of
select="//rap:RapSheet/rap:Introduction/rap:Caveat/j:CaveatReferenceDate"/>
      </xsl:if>
    <xsl:text>)</xsl:text>
  </xsl:variable>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text" select="$CaveatData"/>
    <xsl:with-param name="StartPos" select="$posCol1"/>
    <xsl:with-param name="EndPos" select="$posMaxWidth"/>
    <xsl:with-param name="WrappedStartPos" select="$posCol1"/>
    <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
    <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
  <xsl:text>#10;</xsl:text>
</xsl:for-each>
<!-- ##### Identification Section ##### -->

```

```

<xsl:text>&#10;***** IDENTIFICATION
*****&#10;&#10;</xsl:text>
<xsl:if test="count(/rap:RapSheet/j:Subject/j:PersonLivingIndicator) > 0">
  <!-- dr: JTF 27-FEB-03: Deceased Date added. -->
  <xsl:text>Deceased </xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"
select="concat(/rap:RapSheet/j:Subject/j:PersonDeathDate,'
(/rap:RapSheet/j:Subject/j:PersonLivingIndicator/@j:reportingOrganizationText,',
'/rap:RapSheet/j:Subject/j:PersonLivingIndicator/@j:reportedDate,',
'/rap:RapSheet/j:Subject/j:PersonLivingIndicator/@j:commentText,')'"/>
    <xsl:with-param name="StartPos" select="$posCol1"/>
    <xsl:with-param name="EndPos" select="$posMaxWidth"/>
    <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
    <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
    <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
  <xsl:text>&#10;</xsl:text>
</xsl:if>
<!-- Edited 05/03 -->
<xsl:if test="count(/rap:RapSheet/j:Subject/j:PersonName) > 0">
  <xsl:text>Subject Name(s)&#10;&#10;</xsl:text>
  <xsl:for-each select="//rap:RapSheet/j:Subject/j:PersonName">
    <!-- Name -->
    <xsl:value-of select="./j:PersonSurName"/>
    <xsl:text>, </xsl:text>
    <xsl:if test="count(/j:PersonPrefixName) > 0">
      <!-- if Prefix -->
      <xsl:value-of select="./j:PersonPrefixName"/>
      <xsl:text/>

```

```

</xsl:if>
<xsl:if test="count(/j:PersonGivenName) > 0">
  <!-- if First Name -->
  <xsl:value-of select="/j:PersonGivenName"/>
  <xsl:text> </xsl:text>
  <xsl:text/>
</xsl:if>
<xsl:if test="count(/j:PersonMiddleName) > 0">
  <!-- if Middle Name -->
  <xsl:value-of select="/j:PersonMiddleName"/>
  <xsl:text> </xsl:text>
  <xsl:text/>
</xsl:if>
<xsl:if test="count(/j:PersonSuffixName) > 0">
  <!-- if Suffix -->
  <xsl:value-of select="/j:PersonSuffixName"/>
  <xsl:text/>
</xsl:if>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
</xsl:if>
<!-- -->
<!-- Edited 05/03 -->
<xsl:for-each select="//rap:RapSheet/j:Subject/j:PersonAlernateName">
  <!-- Name -->
  <xsl:value-of select="/j:PersonSurName"/>
  <xsl:text>, </xsl:text>
  <xsl:if test="count(/j:PersonPrefixName) > 0">
    <!-- if Prefix -->
    <xsl:value-of select="/j:PersonPrefixName"/>
    <xsl:text/>

```

```

</xsl:if>
<xsl:if test="count(/j:PersonGivenName) > 0">
  <!-- if First Name -->
  <xsl:value-of select="/j:PersonGivenName"/>
  <xsl:text/>
  <xsl:text/>
</xsl:if>
<xsl:if test="count(/j:PersonMiddleName) > 0">
  <!-- if Middle Name -->
  <xsl:value-of select="/j:PersonMiddleName"/>
  <xsl:text/>
  <xsl:text/>
</xsl:if>
<xsl:if test="count(/j:PersonSuffixName) > 0">
  <!-- if Suffix -->
  <xsl:value-of select="/j:PersonSuffixName"/>
  <xsl:text/>
</xsl:if>
<!-- if Not Primary name -->
<xsl:text> (AKA)</xsl:text>
<xsl:text>&#10;</xsl:text>
</xsl:for-each>
<!-- -->
<xsl:text>&#10;Subject Description&#10;</xsl:text>
<!-- FBI Number/State ID Number/DOC Number -->
<!-- Edited 05/03 -->
<xsl:choose>
  <xsl:when
test="count(/rap:RapSheet/j:Subject/j:PersonAssignedIDDDetails/j:PersonFBIID/j:ID) &gt; 0">
    <xsl:choose>

```

```

                                <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonStateID/j:ID) &gt; 0">
                                <xsl:choose>
                                    <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonOtherID/j:IDTypeText = 'Correctional ID'">
                                        <xsl:text>#10;FBI Number          State Id
Number      DOC Number#10;</xsl:text>
                                </xsl:when>
                                <xsl:otherwise>
                                        <xsl:text>#10;FBI Number          State Id
Number      #10;</xsl:text>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:when>
                                <xsl:otherwise>
                                        <xsl:choose>
                                            <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonOtherID/j:IDTypeText = 'Correctional ID'">
                                                <xsl:text>#10;FBI Number
DOC Number#10;</xsl:text>
                                </xsl:when>
                                <xsl:otherwise>
                                        <xsl:text>#10;FBI Number
#10;</xsl:text>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:when>
                                <xsl:otherwise>
                                        <xsl:choose>

```

```

                                <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonStateID/j:ID) &gt; 0">
                                <xsl:choose>
                                    <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonOtherID/j:IDTypeText = 'Correctional ID'">
                                        <xsl:text>#10;
State Id Number
DOC Number#10;</xsl:text>
                                </xsl:when>
                                <xsl:otherwise>
                                        <xsl:text>#10;
State Id Number
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:when>
                                <xsl:otherwise>
                                        <xsl:choose>
                                            <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonOtherID/j:IDTypeText = 'Correctional ID'">
                                                <xsl:text>#10;
DOC
Number#10;</xsl:text>
                                            </xsl:when>
                                            <xsl:otherwise/>
                                        </xsl:choose>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:choose>
                                </xsl:choose>
                                <!-- -->
                                <xsl:for-each select="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonFBIID/j:ID">
                                    <xsl:variable name="i" select="position()"/>
                                    <xsl:value-of select="."/>

```



```

length(../j:PersonStateID[$i]/j:ID) &gt; 0">
  <xsl:call-template name="spaceover">
    <xsl:with-param name="amount" select="$posCol2 - (string-
length(../j:PersonStateID[$i]/j:IDIssuingAuthorityText"/>
    <xsl:with-param name="amountWrote" select="0"/>
  </xsl:call-template>
  <xsl:value-of select="../j:PersonStateID[$i]/j:ID"/>
  <xsl:if test="../j:PersonStateID[$i]/j:IDIssuingAuthorityText">
    <xsl:text> (</xsl:text>
    <xsl:value-of
select="../j:PersonStateID[$i]/j:IDIssuingAuthorityText"/>
    <xsl:text>)</xsl:text>
  </xsl:if>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="spaceover">
    <xsl:with-param name="amount" select="$posCol2 - string-
length(../j:PersonStateID[$i]/j:ID) + 5)"/>
    <xsl:with-param name="amountWrote" select="0"/>
  </xsl:call-template>
</xsl:otherwise>
</xsl:choose>
<xsl:choose>
  <!-- if there is a [column 3] at this index write it, else return -->
  <xsl:when test="count(../j:PersonOtherID[$i]/j:ID) &gt; 0">
    <xsl:if test="../j:PersonOtherID[$i]/j:IDTypeText = 'Correctional ID'">
      <!-- only do if Correctional ID -->
      <xsl:variable name="adj" select="$posCol3 - ($posCol2 + (string-
length(../j:PersonStateID/j:ID) + 5))"/>

```

```

select="../../j:PersonOtherID[$i]/j:ID"/>

select="$posMaxWidth"/>

select="$posCol3"/>

select="$posMaxWidth + 1"/>

select="$posMaxWidth + 1"/>

<xsl:call-template name="wrapIn">
  <xsl:with-param name="Text"

  <xsl:with-param name="StartPos" select="$adj"/>
  <xsl:with-param name="EndPos"

  <xsl:with-param name="WrapedStartPos"

  <xsl:with-param name="CurrentPos"

  <xsl:with-param name="WrapWidth"

  <xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
</xsl:if>
</xsl:when>
<xsl:otherwise/>
</xsl:choose>
<xsl:if test="count(.) &lt;= count ../../j:PersonStateID/j:ID) or count(.) &lt;=
count ../../j:PersonOtherID/j:ID)">
  <xsl:text>#10;</xsl:text>
</xsl:if>
</xsl:for-each>
<!-- Check to see if there are more [column 2] or [column 3] nodes than [column 1] and write
them if so -->
<xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonStateID/j:ID)
&gt; count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonFBIID/j:ID)">
  <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonStateID/j:ID">
  <xsl:variable name="i" select="position()"/>
  <xsl:if test="$i &gt; count ../../j:PersonFBIID/j:ID)">

```

```

        <xsl:call-template name="spaceover">
            <xsl:with-param name="amount" select="$posCol2"/>
            <xsl:with-param name="amountWrote" select="0"/>
        </xsl:call-template>
        <xsl:value-of select="."/>
        <xsl:if test="../../j:PersonStateID[$i]/j:IDIssuingAuthorityText">
            <xsl:text> (</xsl:text>
            <xsl:value-of
select="../../j:PersonStateID[$i]/j:IDIssuingAuthorityText"/>
            <xsl:text>)</xsl:text>
        </xsl:if>
        <xsl:if test="count(../../j:PersonOtherID[$i]/j:ID) > 0">
            <!-- check to see if there is a [column 3] at the current index and
write it if so -->
            <xsl:call-template name="spaceover">
                <xsl:with-param name="amount" select="$posCol3 -
($posCol2 + string-length(.) + 5)"/>
                <!-- the +13 is to account refDate -->
                <xsl:with-param name="amountWrote" select="0"/>
            </xsl:call-template>
            <xsl:if test="../../j:PersonOtherID[$i]/j:IDTypeText = 'Correctional
ID">
                <xsl:value-of select="../../j:PersonOtherID[$i]/j:ID"/>
            </xsl:if>
        </xsl:if>
        <xsl:text>&#10;</xsl:text>
    </xsl:if>
</xsl:for-each>
</xsl:if>
<!-- Check to see if there are stand alone [column 3]'s and write them if so -->

```

```

        <xsl:if
test="count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonOtherID/j:IDTypeText) &gt;
count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonStateID/j:ID)">
        <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonOtherID/j:IDTypeText">
        <xsl:variable name="i" select="position()"/>
        <xsl:if test="$i &gt; count(..../j:PersonStateID) and $i &gt;
count(..../j:PersonFBIID/j:ID)">
                <xsl:call-template name="spaceover">
        <xsl:with-param name="amount" select="$posCol3"/>
        <xsl:with-param name="amountWrote" select="0"/>
        </xsl:call-template>
        <xsl:if test="j:IDTypeText = 'Correctional ID'">
        <xsl:value-of select="j:ID"/>
        </xsl:if>
        <xsl:text>&#10;</xsl:text>
        </xsl:if>
        </xsl:for-each>
        </xsl:if>
        <xsl:text>&#10;</xsl:text>
        <!-- SSN/Driver's License Numbers -->
        <!-- Edited 05/03 -->
        <xsl:choose>
        <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonSSNID) &gt; 0">
        <xsl:choose>
        <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonDriverLicenseID) &gt; 0">
        <xsl:text>Social Security Number Driver's License
Number&#10;</xsl:text>
        </xsl:when>

```

```

                <xsl:otherwise>
                    <xsl:text>Social Security Number          &#10;</xsl:text>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:when>
        <xsl:otherwise>
            <xsl:choose>
                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonDriverLicenseID) &gt; 0">
                    <xsl:text>          Driver's License
Number&#10;</xsl:text>
                </xsl:when>
                <xsl:otherwise/>
            </xsl:choose>
        </xsl:otherwise>
    </xsl:choose>
    <!-- -->
    <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonSSNID/j:ID">
        <xsl:variable name="curPos" select="position()"/>
        <xsl:value-of select="."/>
        <xsl:choose>
            <xsl:when test="count(..../j:PersonDriverLicenseID/j:ID) &gt; 0">
                <xsl:call-template name="spaceover">
                    <xsl:with-param name="amount" select="$posCol2 - string-
length(.)"/>
                    <xsl:with-param name="amountWrote" select="0"/>
                </xsl:call-template>
            </xsl:choose>
            <xsl:when test="count(..../j:PersonDriverLicenseID[position() =
$curPos]) &gt; 0">

```

```

                                <!-- if there is a dlNumber at the same index write it. -->
                                <xsl:value-of
select=" ../../j:PersonDriverLicenseID[$curPos]/j:ID"/>
                                <xsl:text> (</xsl:text>
                                <xsl:value-of
select=" ../../j:PersonDriverLicenseID[$curPos]/j:IDIssuingAuthorityText"/>
                                <xsl:text>)&#10;</xsl:text>
                                </xsl:when>
                                <xsl:otherwise>
                                    <xsl:text>&#10;</xsl:text>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:when>
                                <xsl:otherwise>
                                    <xsl:text>&#10;</xsl:text>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:for-each>
                                <xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonSSNID) &gt;
0">
                                    <xsl:text>&#10;</xsl:text>
                                </xsl:if>
                                <xsl:if
test="count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonDriverLicenseID) &gt;
count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonSSNID)">
                                    <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonAssignedIDDetails/j:PersonDriverLicenseID">
                                        <xsl:if test="position() &gt; count(..j:PersonSSNID)">
                                            <xsl:call-template name="spaceover">
                                                <xsl:with-param name="amount" select="$posCol2"/>

```

```

        <xsl:with-param name="amountWrote" select="0"/>
    </xsl:call-template>
    <xsl:value-of select="j:ID"/>
    <xsl:text> (</xsl:text>
    <xsl:value-of select="j:IDIssuingAuthorityText"/>
    <xsl:text>)</xsl:text>
    <xsl:text>#10;</xsl:text>
    </xsl:if>
</xsl:for-each>
</xsl:if>

<!-- Miscellaneous Numbers -->
<!-- Edited 05/03 -->
<xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonAssignedIDDDetails/j:PersonOtherID) &gt; 0
and //rap:RapSheet/j:Subject/j:PersonAssignedIDDDetails/j:PersonOtherID/j:IDTypeText!='Correctional ID'">
    <xsl:text>Miscellaneous Numbers#10;</xsl:text>
</xsl:if>
<!-- -->
<xsl:for-each select="//rap:RapSheet/j:Subject/j:PersonAssignedIDDDetails/j:PersonOtherID">
    <xsl:if test="j:IDTypeText != 'Correctional ID'">
        <xsl:value-of select="j:ID"/>
        <xsl:call-template name="spaceover">
            <xsl:with-param name="amount" select="$posCol2 - (string-length(j:ID)
+ 1)"/>
            <xsl:with-param name="amountWrote" select="0"/>
        </xsl:call-template>
        <xsl:value-of select="j:IDTypeText"/>
        <xsl:call-template name="spaceover">
            <xsl:with-param name="amount" select="$posCol3 - ($posCol2 + string-
length(j:IDTypeText))"/>
            <xsl:with-param name="amountWrote" select="0"/>

```

```

        </xsl:call-template>
        <xsl:value-of select="j:IDIssuingAuthorityText"/>
        <xsl:text>#10;</xsl:text>
    </xsl:if>
</xsl:for-each>
<!-- Sex/Race/Skin Tone -->
<!-- Edited 05/03 -->
<xsl:choose>
    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSexText) > 0">
        <xsl:choose>
            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonRaceText) > 0">
                <xsl:choose>
                    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText) > 0">
                        <xsl:text>#10;Sex           Race
Skin Tone#10;</xsl:text>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:text>#10;Sex           Race
#10;</xsl:text>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:when>
            <xsl:otherwise>
                <xsl:choose>
                    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText) > 0">
                        <xsl:text>#10;Sex
Skin Tone#10;</xsl:text>
                    </xsl:when>
                </xsl:choose>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
        <xsl:choose>
            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText) > 0">
                <xsl:text>#10;Sex
Skin Tone#10;</xsl:text>
            </xsl:when>
        </xsl:choose>
    </xsl:otherwise>
</xsl:choose>

```



```

                                </xsl:when>
                                <xsl:otherwise>
                                    <xsl:text>#10;Sex
#10;</xsl:text>
                                </xsl:otherwise>
                            </xsl:choose>
                        </xsl:otherwise>
                    </xsl:choose>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:choose>
                        <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonRaceText) > 0">
                            <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText) > 0">
                                    <xsl:text>#10;
Skin Tone#10;</xsl:text>
                                </xsl:when>
                                <xsl:otherwise>
                                    <xsl:text>#10;
#10;</xsl:text>
                                </xsl:otherwise>
                            </xsl:choose>
                        </xsl:when>
                        <xsl:otherwise>
                            <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText) > 0">
                                    <xsl:text>#10;
Tone#10;</xsl:text>
                                </xsl:when>
                                <xsl:otherwise>
                                    <xsl:text>#10;
Skin

```

```

                </xsl:when>
                <xsl:otherwise/>
            </xsl:choose>
        </xsl:otherwise>
    </xsl:choose>
</xsl:choose>
<!-- -->
<xsl:for-each select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSexText">
    <xsl:variable name="i" select="position()"/>
    <xsl:value-of select="."/>
    <xsl:choose>
        <!-- if there is a race at this index write it, else spaceover -->
        <xsl:when test="count(..j:PersonRaceText[$i]) &gt; 0">
            <xsl:call-template name="spaceover">
                <xsl:with-param name="amount" select="$posCol2 - string-
length()"/>
                <xsl:with-param name="amountWrote" select="0"/>
            </xsl:call-template>
            <xsl:value-of select="../j:PersonRaceText[$i]"/>
        </xsl:when>
        <xsl:otherwise>
            <xsl:call-template name="spaceover">
                <xsl:with-param name="amount" select="$posCol2 - string-
length()"/>
                <xsl:with-param name="amountWrote" select="0"/>
            </xsl:call-template>
        </xsl:otherwise>
    </xsl:choose>
</xsl:choose>
<!-- if there is a skin tone at this index write it, else returnr -->

```

```

        <xsl:when test="count(..j:PersonSkinToneText[$i]) &gt; 0">
            <xsl:call-template name="spaceover">
                <xsl:with-param name="amount" select="$posCol3 - ($posCol2 +
string-length(..j:PersonRaceText[$i]))"/>
                <xsl:with-param name="amountWrote" select="0"/>
            </xsl:call-template>
            <xsl:value-of select="..j:PersonSkinToneText[$i]"/>
        </xsl:when>
        <xsl:otherwise>
            <!-- xsl:text>&#10;</xsl:text -->
        </xsl:otherwise>
    </xsl:choose>
    <xsl:if test="count(.) &lt;= count(..j:PersonRaceText) or count(.) &lt;=
count(..j:PersonSkinToneText)">
        <xsl:text>&#10;</xsl:text>
    </xsl:if>
</xsl:for-each>
<!-- Check to see if there are more race or skinTone nodes than sex and write them if so -->
<xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonRaceText) &gt;
count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSexText)">
    <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonRaceText">
        <xsl:variable name="i" select="position()"/>
        <xsl:if test="$i &gt; count(..j:PersonSexText
)">
            <xsl:call-template name="spaceover">
                <xsl:with-param name="amount" select="$posCol2"/>
                <xsl:with-param name="amountWrote" select="0"/>
            </xsl:call-template>
            <xsl:value-of select="."/>
            <xsl:if test="count(..j:PersonSkinToneText[$i]) &gt; 0">

```

```

it if so -->
($posCol2 + string-length(.))"/>
<!-- check to see if there is skinTone at the current index and write
<xsl:call-template name="spaceover">
  <xsl:with-param name="amount" select="$posCol3 -
    <xsl:with-param name="amountWrote" select="0"/>
  </xsl:call-template>
  <xsl:value-of select="../j:PersonSkinToneText[$i]"/>
</xsl:if>
  <xsl:text>#10;</xsl:text>
</xsl:if>
  <!-- xsl:text>#10;x</xsl:text -->
  <!-- moved up in the if -->
</xsl:for-each>
</xsl:if>
<!-- Check to see if there are more skinTones (stand alone) than race and write them if so -->
<xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText)
  &gt; count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonRaceText)">
  <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText">
  <xsl:variable name="i" select="position()"/>
  <xsl:if test="$i &gt; count(../j:PersonRaceText) and $i &gt;
count(../j:PersonSexText)">
    <xsl:call-template name="spaceover">
      <xsl:with-param name="amount" select="$posCol3"/>
      <xsl:with-param name="amountWrote" select="0"/>
    </xsl:call-template>
    <xsl:value-of select="."/>
    <xsl:text>#10;</xsl:text>
  </xsl:if>
</xsl:for-each>

```

```

        </xsl:if>
        <!-- Edited 05/03 -->
        <xsl:choose>
            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSexText) &gt; 0">
                <xsl:choose>
                    <xsl:when test="count(
//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonRaceText) &gt; 0">
                        <xsl:choose>
                            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText) &gt; 0"/>
                                <xsl:otherwise/>
                            </xsl:choose>
                        </xsl:when>
                        <xsl:otherwise>
                            <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText) &gt; 0"/>
                                    <xsl:otherwise>
                                        <xsl:text>&#10;</xsl:text>
                                    </xsl:otherwise>
                                </xsl:choose>
                            </xsl:otherwise>
                        </xsl:choose>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:choose>
                            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonRaceText) &gt; 0">
                                <xsl:choose>

```

```

                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText) &gt; 0"/>
                                <xsl:otherwise/>
                                </xsl:choose>
                                </xsl:when>
                                <xsl:otherwise>
                                <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonSkinToneText) &gt; 0"/>
                                <xsl:otherwise/>
                                </xsl:choose>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:otherwise>
                                </xsl:choose>
                                <!-- -->
                                <!-- Height/Weight/DOB -->
                                <!-- Edited 05/03 -->
                                <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHeightMeasure) &gt; 0">
                                <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonWeightMeasure) &gt; 0">
                                <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonBirthDate) &gt; 0">
                                <xsl:text>#10;Height
                                Weight
Date of Birth#10;</xsl:text>
                                </xsl:when>
                                <xsl:otherwise>

```

&#10;</xsl:text>	<xsl:text>&#10;Height	Weight
	</xsl:otherwise> </xsl:choose> </xsl:when> <xsl:otherwise> <xsl:choose> <xsl:when	
test="count(//rap:RapSheet/j:Subject/j:PersonBirthDate) &gt; 0">	<xsl:text>&#10;Height	
Date of Birth&#10;</xsl:text>	</xsl:when> <xsl:otherwise> <xsl:text>&#10;Height	
&#10;</xsl:text>	</xsl:otherwise> </xsl:choose> </xsl:otherwise> </xsl:choose> </xsl:when> <xsl:otherwise> <xsl:choose> <xsl:when	
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonWeightMeasure) &gt; 0">	<xsl:choose> <xsl:when	
test="count(//rap:RapSheet/j:Subject/j:PersonBirthDate) &gt; 0">	<xsl:text>&#10;	Weight
Date of Birth&#10;</xsl:text>	</xsl:when> <xsl:otherwise>	

```

                                <xsl:text>#10;
                                Weight
#10;</xsl:text>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:when>
                                <xsl:otherwise>
                                <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonBirthDate) > 0">
                                <xsl:text>#10;
                                Date
of Birth#10;</xsl:text>
                                </xsl:when>
                                <xsl:otherwise/>
                                </xsl:choose>
                                </xsl:otherwise>
                                </xsl:choose>
                                </xsl:otherwise>
                                </xsl:choose>
                                <!-- -->
                                <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHeightMeasure">
                                <xsl:variable name="i" select="position()"/>
                                <xsl:variable name="HeightData">
                                <xsl:choose>
                                <xsl:when test="@j:personHeightUnitCode = 'ncic'">
                                <xsl:value-of select="substring(.,1,1)"/>
                                <xsl:text>'</xsl:text>
                                <xsl:value-of select="substring(.,2,2)"/>
                                <xsl:text>"</xsl:text>
                                <xsl:if
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHeightMeasure/@j:reportedDate) > 0">

```



```

                <xsl:text> (</xsl:text>
                <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHeightMeasure/@j:reportedDate"/>
                <xsl:text>)</xsl:text>
            </xsl:if>
        </xsl:when>
        <xsl:otherwise>
            <xsl:value-of select="."/>
            <xsl:value-of select="@j:personHeightUnitCode"/>
            <xsl:text> (</xsl:text>
            <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHeightMeasure/@j:reportedDate">
                <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHeightMeasure/@j:reportedDate"/>
                </xsl:if>
                <xsl:text>)</xsl:text>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:value-of select="$HeightData"/>
    <xsl:choose>
        <!-- if there is a [column 2] at this index write it, else spaceover -->
        <xsl:when test="count(..j:PersonWeightMeasure[$i]) > 0">
            <xsl:call-template name="spaceover">
                <xsl:with-param name="amount" select="$posCol2 - (string-
length($HeightData))"/>
                <xsl:with-param name="amountWrote" select="0"/>
            </xsl:call-template>
        <xsl:value-of select="../j:PersonWeightMeasure[$i]"/>
        <xsl:text/>
    </xsl:choose>

```

```

                                <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHeightMeasure/@j:reportedDate">
                                <xsl:text> (</xsl:text>
                                <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHeightMeasure/@j:reportedDate"/>
                                <xsl:text>)</xsl:text>
                                </xsl:if>
                                </xsl:when>
                                <xsl:otherwise>
                                <xsl:call-template name="spaceover">
                                <xsl:with-param name="amount" select="$posCol2 - string-
length(.)-6"/>
                                <xsl:with-param name="amountWrote" select="0"/>
                                </xsl:call-template>
                                </xsl:otherwise>
                                </xsl:choose>
                                <xsl:choose>
                                <!-- if there is a [column 3] at this index write it, else return -->
                                <xsl:when test="count(//rap:RapSheet/j:Subject/j:PersonBirthDate[ $i]) &gt; 0">
                                <xsl:choose>
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonWeightMeasure/@j:reportedDate) &gt; 0">
                                <xsl:call-template name="spaceover">
                                <xsl:with-param name="amount" select="$posCol3
- ($posCol2 + string-length(..j:PersonWeightMeasure[ $i]) + 13)"/>
                                <!-- the +20 is to account for pounds and refDate --
                                <xsl:with-param name="amountWrote"
select="0"/>
                                </xsl:call-template>
                                </xsl:when>

```

```

                                <xsl:otherwise>
                                    <xsl:call-template name="spaceover">
                                        <xsl:with-param name="amount" select="$posCol3
- ($posCol2 + string-length(..:/j:PersonWeightMeasure[$i]) + 4)"/>
                                        <!-- the +20 is to account for pounds and refDate --
>
                                        <xsl:with-param name="amountWrote"
select="0"/>
                                </xsl:call-template>
                                </xsl:otherwise>
                            </xsl:choose>
                            <xsl:value-of select="//rap:RapSheet/j:Subject/j:PersonBirthDate[$i]"/>
                            <xsl:when>
                                <xsl:otherwise/>
                            </xsl:choose>
                            <xsl:if test="count(.) &lt;= count(..:/j:PersonWeightMeasure) or count(.) &lt;=
count(//rap:RapSheet/j:Subject/j:PersonBirthDate)">
                                <xsl:text>&#10;</xsl:text>
                            </xsl:if>
                        </xsl:for-each>
                        <!-- Check to see if there are more [column 2] or [column 3] nodes than [column 1] and write
them if so -->
                            <xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonWeightMeasure)
&gt; count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHeightMeasure)">
                                <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonWeightMeasure">
                                    <xsl:variable name="i" select="position()"/>
                                    <xsl:if test="$i &gt; count(..:/j:PersonHeightMeasure)">
                                        <xsl:call-template name="spaceover">
                                            <xsl:with-param name="amount" select="$posCol2"/>
                                            <xsl:with-param name="amountWrote" select="0"/>

```

```

        </xsl:call-template>
        <xsl:value-of select="."/>
        <xsl:text/>
        <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonWeightMeasure/@j:reportedDate">
            <xsl:text> (</xsl:text>
            <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonWeightMeasure/@j:reportedDate"/>
            <xsl:text>)</xsl:text>
        </xsl:if>
        <xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonBirthDate[$i]) &gt;
0">
            <!-- check to see if there is a [column 3] at the current index and
write it if so -->
            <xsl:choose>
                <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonWeightMeasure/@j:reportedDate">
                    <xsl:call-template name="spaceover">
                        <xsl:with-param name="amount"
select="$posCol3 - ($posCol2 + string-length(.) + 20)"/>
                        <!-- the +20 is to account for pounds and
refDate -->
                        <xsl:with-param name="amountWrote"
select="0"/>
                    </xsl:call-template>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:call-template name="spaceover">
                        <xsl:with-param name="amount"
select="$posCol3 - ($posCol2 + string-length(.) + 8)"/>

```



```

        <xsl:choose>
            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHeightMeasure) &gt; 0">
                <xsl:choose>
                    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonWeightMeasure) &gt; 0">
                        <xsl:choose>
                            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonBirthDate) &gt; 0"/>
                                <xsl:otherwise/>
                            </xsl:choose>
                        </xsl:when>
                    <xsl:otherwise>
                        <xsl:choose>
                            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonBirthDate) &gt; 0"/>
                                <xsl:otherwise>
                                    <xsl:text>#10;</xsl:text>
                                </xsl:otherwise>
                            </xsl:choose>
                        </xsl:otherwise>
                    </xsl:choose>
                </xsl:when>
            <xsl:otherwise/>
        </xsl:choose>
<!-- -->
<!-- Hair Color/Eye Color/Fingerprint Patern -->
<!-- Edited 05/03 -->
<xsl:choose>
    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHairColorText) &gt; 0">

```

```

        <xsl:choose>
            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText) &gt; 0">
                <xsl:choose>
                    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricValueText) &gt; 0">
                        <xsl:text>#10;Hair Color           Eye Color
Fingerprint Pattern#10;</xsl:text>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:text>#10;Hair Color           Eye Color
#10;</xsl:text>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:when>
            <xsl:otherwise>
                <xsl:choose>
                    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricValueText) &gt; 0">
                        <xsl:text>#10;Hair Color
Fingerprint Pattern#10;</xsl:text>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:text>#10;Hair Color
#10;</xsl:text>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:when>
<xsl:otherwise>

```

```

        <xsl:choose>
            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText) > 0">
                <xsl:choose>
                    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricValueText) > 0">
                        <xsl:text>#10;           Eye Color
Fingerprint Pattern#10;</xsl:text>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:text>#10;           Eye Color
#10;</xsl:text>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:when>
            <xsl:otherwise>
                <xsl:choose>
                    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricValueText) > 0">
                        <xsl:text>#10;
Fingerprint Pattern#10;</xsl:text>
                    </xsl:when>
                    <xsl:otherwise/>
                </xsl:choose>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:choose>
<!-- -->
<xsl:for-each select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHairColorText">
    <xsl:variable name="i" select="position()"/>

```



```

        <xsl:value-of select="."/>
        <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHairColorText/@j:reportedDate">
            <xsl:text> (</xsl:text>
            <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHairColorText/@j:reportedDate"/>
            <xsl:text>)</xsl:text>
        </xsl:if>
        <xsl:choose>
            <!-- if there is a [column 2] at this index write it, else spaceover -->
            <xsl:when test="count(..j:PersonEyeColorText[$i]) > 0">
                <xsl:choose>
                    <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHairColorText/@j:reportedDate">
                        <xsl:call-template name="spaceover">
                            <xsl:with-param name="amount" select="$posCol2
- (string-length(.) + 13)"/>
                            <xsl:with-param name="amountWrote"
select="0"/>
                        </xsl:call-template>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:call-template name="spaceover">
                            <xsl:with-param name="amount" select="$posCol2
- (string-length(.))"/>
                            <xsl:with-param name="amountWrote"
select="0"/>
                        </xsl:call-template>
                    </xsl:otherwise>
                </xsl:choose>
            <xsl:value-of select="../j:PersonEyeColorText[$i]"/>

```

```

                                <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText/@j:reportedDate">
                                <xsl:text> (</xsl:text>
                                <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText/@j:reportedDate"/>
                                <xsl:text>)</xsl:text>
                                </xsl:if>
                                </xsl:when>
                                <xsl:otherwise>
                                <xsl:call-template name="spaceover">
                                <xsl:with-param name="amount" select="$posCol2 - string-
length(.)"/>
                                <xsl:with-param name="amountWrote" select="0"/>
                                </xsl:call-template>
                                </xsl:otherwise>
                                </xsl:choose>
                                <xsl:choose>
                                <!-- if there is a [column 3] at this index write it, else return -->
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricValueText[$i]) > 0">
                                <xsl:choose>
                                <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText/@j:reportedDate">
                                <xsl:call-template name="spaceover">
                                <xsl:with-param name="amount" select="$posCol3
- ($posCol2 + string-length(..j:PersonEyeColorText[$i]) + 13)"/>
                                <!-- the +13 is to account for refDate -->
                                <xsl:with-param name="amountWrote"
select="0"/>
                                </xsl:call-template>
                                </xsl:when>

```

```

                                <xsl:otherwise>
                                    <xsl:call-template name="spaceover">
                                        <xsl:with-param name="amount" select="$posCol3
- ($posCol2 + string-length(..:/j:PersonEyeColorText[$i]))"/>
                                        <xsl:with-param name="amountWrote"
select="0"/>
                                    </xsl:call-template>
                                </xsl:otherwise>
                            </xsl:choose>
                            <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricValueText[$i]"/>
                            <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricEncodingMethodText[$i]">
                                <xsl:value-of select="concat('
(../rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricEncodingMethodText[$i],)')"/>
                                </xsl:if>
                            </xsl:when>
                            <xsl:otherwise>
                                <!-- xsl:text>#10;</xsl:text -->
                            </xsl:otherwise>
                        </xsl:choose>
                        <xsl:if test="count(.) &lt;= count(..:/j:PersonEyeColorText) or count(.) &lt;=
count(../rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricEncodingMethodText)">
                            <xsl:text>#10;</xsl:text>
                        </xsl:if>
                    </xsl:for-each>
                    <!-- Check to see if there are more [column 2] or [column 3] nodes than [column 1] and write
them if so -->
                    <xsl:if test="count(../rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText) &gt;
count(../rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonHairColorText)">

```

```

        <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText">
            <xsl:variable name="i" select="position()"/>
            <xsl:if test="$i > count(..j:PersonHairColorText)">
                <xsl:call-template name="spaceover">
                    <xsl:with-param name="amount" select="$posCol2"/>
                    <xsl:with-param name="amountWrote" select="0"/>
                </xsl:call-template>
                <xsl:value-of select="."/>
                <xsl:text> (</xsl:text>
                <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText/@j:reportedDate"/>
                <xsl:text></xsl:text>
                <xsl:if test="count(..j:PersonFingerPrintSet[$i]) > 0">
                    <!-- check to see if there is a [column 3] at the current index and
write it if so -->
                    <xsl:choose>
                        <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText/@j:reportedDate) > 0">
                            <xsl:call-template name="spaceover">
                                <xsl:with-param name="amount"
select="$posCol3 - ($posCol2 + string-length(..j:PersonEyeColorText[$i]) + 13)"/>
                                    <!-- the +13 is to account for refDate -->
                                <xsl:with-param name="amountWrote"
select="0"/>
                                    </xsl:call-template>
                            </xsl:when>
                            <xsl:otherwise>
                                <xsl:call-template name="spaceover">
                                    <xsl:with-param name="amount"
select="$posCol3 - ($posCol2 + string-length(..j:PersonEyeColorText[$i]) + 13)"/>
                                </xsl:call-template>
                            </xsl:otherwise>
                        </xsl:choose>
                    </xsl:if>
                </xsl:value-of>
            </xsl:if>
        </xsl:for-each>

```

```

refDate -->
select="0"/>
<!-- the +20 is to account for pounds and
<xsl:with-param name="amountWrote"
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
<xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricValueText[$i]"/>
<xsl:if
test="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricEncodingMethodText[$i]">
<xsl:value-of select="concat('
(, //rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricEncodingMethodText[$i],)')"/>
</xsl:if>
</xsl:if>
<xsl:text>#10;</xsl:text>
</xsl:if>
</xsl:for-each>
</xsl:if>
<!-- Check to see if there are stand alone [column 3]'s and write them if so -->
<xsl:if
test="count(//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricValueText) >
count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText)">
<xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricValueText">
<xsl:variable name="i" select="position()"/>
<xsl:if test="$i > count(..j:PersonEyeColorText) and $i >
count(..j:PersonHairColorText)">
<xsl:choose>
<xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonEyeColorText/@j:reportedDate) > 0">

```

```

- ($posCol2 + string-length(..:/j:PersonEyeColorText[$i]) + 13)"/>
<xsl:call-template name="spaceover">
  <xsl:with-param name="amount" select="$posCol3
  <!-- the +13 is to account for refDate -->
  <xsl:with-param name="amountWrote"
    </xsl:call-template>
  </xsl:when>
  <xsl:otherwise>
    <xsl:call-template name="spaceover">
      <xsl:with-param name="amount" select="$posCol3
      <!-- the +20 is to account for pounds and refDate --
      <xsl:with-param name="amountWrote"
    </xsl:call-template>
  </xsl:otherwise>
  </xsl:choose>
  <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricValueText[$i]"/>
  <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricEncodingMethodText[$i]">
    <xsl:value-of select="concat(
(../rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:BiometricEncodingMethodText[$i],')')"/>
    </xsl:if>
    <xsl:text>#10;</xsl:text>
  </xsl:if>
  </xsl:for-each>
</xsl:if>
<!-- Scars Marks and Tattoos -->

```

```

&gt; 0">
    <xsl:if test="count(/rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonPhysicalFeature)
        <!-- Edited 05/03 -->
        <xsl:text>#10;Scars, Marks, and Tattoos#10;</xsl:text>
        <!-- -->
        <xsl:text>Code          Description, Comments, and Images#10;</xsl:text>
    </xsl:if>
    <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonPhysicalFeature">
        <!-- get SMT Code formatted properly and return into a variable -->
        <xsl:variable name="SMTCode">
            <xsl:call-template name="doSMTCode">
                <xsl:with-param name="codeSource"
select="./j:PhysicalFeatureTypeCode"/>
            </xsl:call-template>
        </xsl:variable>
        <xsl:value-of select="$SMTCode"/>

        <!-- space over the correct amount to start the next column -->
        <xsl:call-template name="spaceover">
            <xsl:with-param name="amount" select="$posCol2 - (string-
length($SMTCode))"/>
            <xsl:with-param name="amountWrote" select="0"/>
        </xsl:call-template>

        <!-- get SMT Description Info -->
        <xsl:choose>
            <xsl:when test="count(/j:PhysicalFeatureDescriptionText) &gt; 0">
                <xsl:variable name="SMTDescription"
select="concat(/j:PhysicalFeatureCategoryText, ', ', /j:PhysicalFeatureDescriptionText)"/>

```

it is all formatted. -->

```
select="$SMTDescription"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="concat(/j:PhysicalFeatureCategoryText, ', ', /j:PhysicalFeatureTypeText)"/>
```

now that it is all formatted. -->

```
select="$SMTDescription"/>
```

```
select="$posCol1"/>
```

```
select="$posMaxWidth"/>
```

<!-- djr(19-mar-03) Write out the SMT Description Info now that

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos" select="$posCol1"/>
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun" select="0"/>
```

```
</xsl:call-template>
```

```
<xsl:text>#10;</xsl:text>
```

```
</xsl:when>
```

```
<xsl:otherwise>
```

```
<xsl:variable name="SMTDescription"
```

```
</xsl:variable>
```

<!-- djr(19-mar-03) Write out the SMT Description Info

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos"
```

```
<xsl:with-param name="EndPos"
```



```

select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
<xsl:with-param name="WrappedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<!-- Blood Type/Medical Conditions -->
<!-- Edited 05/03 -->
<xsl:choose>
<xsl:when
test="count(/rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonBloodTypeText) > 0">
<xsl:choose>
<xsl:when
test="count(/rap:RapSheet/j:Subject/j:PersonMedicalDetails/j:PersonMedicalCondition/j:MedicalConditionText) > 0">
<xsl:text>#10;Blood Type Medical
Condition#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:text>#10;Blood Type #10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
<xsl:choose>

```

```

                                <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonMedicalDetails/j:PersonMedicalCondition/j:MedicalConditionText) &gt; 0">
                                <xsl:text>#10;           Medical
Condition#10;</xsl:text>
                                </xsl:when>
                                <xsl:otherwise/>
                                </xsl:choose>
                                <xsl:otherwise>
</xsl:choose>
                                <!-- -->
                                <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonBloodTypeText">
                                <xsl:variable name="curPos" select="position()"/>
                                <xsl:value-of select="."/>
                                <xsl:choose>
                                <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonMedicalDetails/j:PersonMedicalCondition/j:MedicalConditionText) &gt; 0">
                                <xsl:call-template name="spaceover">
                                <xsl:with-param name="amount" select="$posCol2 - string-
length(.)"/>
                                <xsl:with-param name="amountWrote" select="0"/>
                                </xsl:call-template>
                                <xsl:choose>
                                <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonMedicalDetails/j:PersonMedicalCondition[position() = $curPos]) &gt; 0">
                                <!-- if there is a medicalCondition at the same index write
it. -->
                                <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonMedicalDetails/j:PersonMedicalCondition/j:MedicalConditionText[$curPos]"/>
                                <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonMedicalDetails/j:PersonMedicalCondition[$curPos]/@j:reportedDate">

```

```

                <xsl:text> (</xsl:text>
                <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonMedicalDetails/j:PersonMedicalCondition[$curPos]/@j:reportedDate"/>
                <xsl:text>)</xsl:text>
            </xsl:if>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>#10;</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
</xsl:when>
<xsl:otherwise>
    <xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<!-- djr(19-mar-03) make sure 2nd column isn't stand alone. -->
<xsl:choose>
    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonMedicalDetails/j:PersonMedicalCondition) >
count(//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonBloodTypeText)">
        <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonMedicalDetails/j:PersonMedicalCondition">
            <xsl:if test="position() > count(//j:PersonBloodTypeText)">
                <xsl:text>#10;</xsl:text>
                <xsl:call-template name="spaceover">
                    <xsl:with-param name="amount" select="$posCol2"/>
                    <xsl:with-param name="amountWrote" select="0"/>
                </xsl:call-template>
                <xsl:value-of select="./j:MedicalConditionText"/>
                <xsl:if test="@j:reportedDate">

```

```

        <xsl:text> (</xsl:text>
        <xsl:value-of select="@j:reportedDate"/>
        <xsl:text>)</xsl:text>
    </xsl:if>
    <xsl:text>#10;</xsl:text>
    </xsl:if>
</xsl:for-each>
<xsl:text>#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
    <xsl:text>#10;#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
<!-- Place of Birth/Citizenship/Ethnicity -->
<!-- Edited 05/03 -->
<xsl:choose>
    <xsl:when test="count(/rap:RapSheet/j:Subject/j:PersonBirthLocation) > 0">
        <xsl:choose>
            <xsl:when
test="count(/rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText) > 0">
                <xsl:choose>
                    <xsl:when
test="count(/rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText) > 0">
                        <xsl:text>Place of Birth      Citizenship
Ethnicity#10;</xsl:text>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:text>Place of Birth      Citizenship
#10;</xsl:text>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:when>
        </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
        <xsl:text>#10;</xsl:text>
    </xsl:otherwise>
</xsl:choose>

```

```

        </xsl:when>
        <xsl:otherwise>
            <xsl:choose>
                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText) > 0">
                    <xsl:text>Place of Birth
Ethnicity&#10;</xsl:text>

```

```

                </xsl:when>
                <xsl:otherwise>
                    <xsl:text>Place of Birth
&#10;</xsl:text>

```

```

            </xsl:otherwise>
        </xsl:choose>
    </xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
    <xsl:choose>
        <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText) > 0">

```

```

            <xsl:choose>
                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText) > 0">
                    <xsl:text>
Ethnicity&#10;</xsl:text>

```

```

                </xsl:when>
                <xsl:otherwise>
                    <xsl:text>
Citizenship

```

```

            </xsl:otherwise>
        </xsl:choose>

```

```

        </xsl:when>
        <xsl:otherwise>
            <xsl:choose>
                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText) > 0">
                    <xsl:text>#10;
Ethnicity#10;</xsl:text>
                </xsl:when>
                <xsl:otherwise/>
            </xsl:choose>
        </xsl:otherwise>
    </xsl:choose>
</xsl:otherwise>
</xsl:choose>
<!-- -->
<xsl:for-each select="//rap:RapSheet/j:Subject/j:PersonBirthLocation">
    <xsl:variable name="i" select="position()"/>
    <xsl:value-of select="./j:LocationName"/>
    <xsl:choose>
        <!-- if there is a [column 2] at this index write it, else spaceover -->
        <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText[$i]) > 0">
            <xsl:call-template name="spaceover">
                <xsl:with-param name="amount" select="$posCol2 - (string-
length(./j:LocationName))"/>
                <xsl:with-param name="amountWrote" select="0"/>
            </xsl:call-template>
            <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText[$i]"/>
        </xsl:if
test="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText[$i]/@j:reportedDate">

```

```

                                <xsl:text> (</xsl:text>
                                <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText[$i]/@j:reportedDate"/>
                                <xsl:text>)</xsl:text>
                                </xsl:if>
                                </xsl:when>
                                <xsl:otherwise>
                                <xsl:call-template name="spaceover">
                                <xsl:with-param name="amount" select="$posCol2 - string-
length(.:j:LocationName)"/>
                                <xsl:with-param name="amountWrote" select="0"/>
                                </xsl:call-template>
                                </xsl:otherwise>
                                </xsl:choose>
                                <xsl:choose>
                                <!-- if there is a [column 3] at this index write it, else return -->
                                <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText[$i]) > 0">
                                <xsl:choose>
                                <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText[$i]/@j:reportedDate">
                                <xsl:call-template name="wrapIn">
                                <xsl:with-param name="Text"
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText[$i]"/>
                                <xsl:with-param name="StartPos"
select="$posCol3 - ($posCol2 + (string-length(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText[$i]) + 13))"/>
                                <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
                                <xsl:with-param name="WrappedStartPos"
select="$posCol3"/>

```

```

select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText[$i]"/>
select="$posCol3 - ($posCol2 + (string-length(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText[$i])))"/>
select="$posMaxWidth"/>
select="$posCol3"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText) or count(.) &lt;=
count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText)">
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<xsl:with-param name="EndPos"
<xsl:with-param name="WrappedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise/>
</xsl:choose>
<xsl:if test="count(.) &lt;=
count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText)">

```



```

        <xsl:text>&#10;</xsl:text>
    </xsl:if>
</xsl:for-each>
<!-- Check to see if there are more [column 2] or [column 3] nodes than [column 1] and write
them if so -->
    <xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText) &gt;
count(//rap:RapSheet/j:Subject/j:PersonBirthLocation)">
        <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText">
            <xsl:variable name="i" select="position()"/>
            <xsl:if test="$i &gt; count(//rap:RapSheet/j:Subject/j:PersonBirthLocation)">
                <xsl:call-template name="spaceover">
                    <xsl:with-param name="amount" select="$posCol2"/>
                    <xsl:with-param name="amountWrote" select="0"/>
                </xsl:call-template>
                <xsl:value-of select="."/>
            <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText/@j:reportedDate">
                <xsl:text> (</xsl:text>
                <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText/@j:reportedDate"/>
                <xsl:text>)</xsl:text>
            </xsl:if>
            <xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonBirthDate[$i]) &gt;
0">
                <!-- check to see if there is a [column 3] at the current index and
write it if so -->
                <xsl:choose>
                    <xsl:when test="@j:reportedDate">
                        <xsl:call-template name="spaceover">

```

```

select="$posCol3 - ($posCol2 + string-length(.) + 13)"/>
select="0"/>
select="$posCol3 - ($posCol2 + string-length(.)"/>
select="0"/>
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText[$i]"/>
count("//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText")>
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText">

```

```

<xsl:with-param name="amount"
<!-- the +13 is to account refDate -->
<xsl:with-param name="amountWrote"
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="spaceover">
<xsl:with-param name="amount"
<!-- the +13 is to account refDate -->
<xsl:with-param name="amountWrote"
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
<xsl:value-of
</xsl:if>
<xsl:text>#10;</xsl:text>
</xsl:if>
</xsl:for-each>
</xsl:if>
<!-- Check to see if there are stand alone [column 3]'s and write them if so -->
<xsl:if test="count("//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText) >
<xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonEthnicityText">
<xsl:variable name="i" select="position()"/>

```

```

        <xsl:if test="$i >
count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonCitizenshipText) and $i >
count(//rap:RapSheet/j:Subject/j:PersonBirthLocation)">
            <xsl:call-template name="spaceover">
                <xsl:with-param name="amount" select="$posCol3"/>
                <xsl:with-param name="amountWrote" select="0"/>
            </xsl:call-template>
            <xsl:value-of select="."/>
            <xsl:text>&#10;</xsl:text>
        </xsl:if>
    </xsl:for-each>
</xsl:if>
<!-- Marital Status/Religion -->
<!-- Edited 05/03 -->
<xsl:choose>
    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonMaritalStatusText) > 0">
        <xsl:choose>
            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonReligionText) > 0">
                <xsl:text>&#10;Marital Status      Religion&#10;</xsl:text>
            </xsl:when>
            <xsl:otherwise>
                <xsl:text>&#10;Marital Status      &#10;</xsl:text>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:when>
    <xsl:otherwise>
        <xsl:choose>
            <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonReligionText) > 0">

```

```

                                <xsl:text>#10;           Religion#10;</xsl:text>
                                </xsl:when>
                                <xsl:otherwise>
                                    <xsl:text>#10;</xsl:text>
                                </xsl:otherwise>
                            </xsl:choose>
                        </xsl:otherwise>
                    </xsl:choose>
                <!-- -->
                <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonMaritalStatusText">
                    <xsl:variable name="curPos" select="position()"/>
                    <xsl:value-of select="."/>
                    <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonMaritalStatusText/@j:reportedDate">
                        <xsl:text> (</xsl:text>
                        <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonMaritalStatusText/@j:reportedDate"/>
                        <xsl:text>)</xsl:text>
                    </xsl:if>
                    <xsl:choose>
                        <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonReligionText) > 0">
                            <xsl:choose>
                                <xsl:when
test="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonReligionText/@j:reportedDate">
                                    <xsl:call-template name="spaceover">
                                        <xsl:with-param name="amount" select="$posCol2
- (string-length(.)+13)"/>
                                        <xsl:with-param name="amountWrote"
select="0"/>

```

```

        </xsl:call-template>
    </xsl:when>
    <xsl:otherwise>
        <xsl:call-template name="spaceover">
            <xsl:with-param name="amount" select="$posCol2

- string-length."/>
            <xsl:with-param name="amountWrote"

select="0"/>
        </xsl:call-template>
    </xsl:otherwise>
</xsl:choose>
<xsl:choose>
    <xsl:when
test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonReligionText[position() = $curPos]) > 0">
        <!-- if there is a religion at the same index write it. -->
        <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonReligionText[$curPos]"/>
        <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonReligionText[$curPos]/@j:reportedDate">
            <xsl:text> (</xsl:text>
            <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonReligionText[$curPos]/@j:reportedDate"/>
            <xsl:text>)</xsl:text>
        </xsl:if>
        <xsl:text>#10;#10;</xsl:text>
    </xsl:when>
    <xsl:otherwise>
        <xsl:text>#10;#10;</xsl:text>
    </xsl:otherwise>
</xsl:choose>
</xsl:when>

```

```

        <xsl:otherwise>
            <xsl:text>#10;#10;</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
</xsl:for-each>
<xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonReligionText) >
count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonMaritalStatusText)">
    <xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonPhysicalDetails/j:PersonReligionText">
        <xsl:if test="position() >
count(//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonMaritalStatusText)">
            <xsl:call-template name="spaceover">
                <xsl:with-param name="amount" select="$posCol2"/>
                <xsl:with-param name="amountWrote" select="0"/>
            </xsl:call-template>
            <xsl:value-of select="."/>
            <xsl:if
test="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonMaritalStatusText/@j:reportedDate">
                <xsl:text> (</xsl:text>
                <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonSocialDetails/j:PersonMaritalStatusText/@j:reportedDate"/>
                <xsl:text>)</xsl:text>
            </xsl:if>
            <xsl:text>#10;</xsl:text>
        </xsl:if>
    </xsl:for-each>
</xsl:if>
<!-- Employment -->
<!-- Edited 05/03 -->
<xsl:choose>
    <xsl:when test="count(//rap:RapSheet/j:Subject/j:Employment) > 0">

```

```

        <xsl:text>Employment&#10;</xsl:text>
    </xsl:when>
    <xsl:otherwise/>
</xsl:choose>
<!-- -->
<xsl:for-each select="//rap:RapSheet/j:Subject/j:Employment">
    <xsl:if test="count(@j:reportedDate) &gt; 0">
        <!-- If there is a reference date print it -->
        <xsl:text>Employment as of </xsl:text>
        <xsl:value-of select="@j:reportedDate"/>
        <xsl:text>&#10;</xsl:text>
    </xsl:if>
    <xsl:text>Occupation</xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text" select="./j:EmploymentOccupationText"/>
        <xsl:with-param name="StartPos" select="$posCol2 - 10"/>
        <xsl:with-param name="EndPos" select="$posMaxWidth"/>
        <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>&#10;</xsl:text>
    <xsl:text>Employer</xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text" select="./j:EmploymentEmployerName"/>
        <xsl:with-param name="StartPos" select="$posCol2 - 8"/>
        <xsl:with-param name="EndPos" select="$posMaxWidth"/>
        <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
    </xsl:call-template>

```

```

        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
    <xsl:if test=".j:EmploymentLocation/j:LocationAddress">
        <xsl:text>Location</xsl:text>
        <xsl:if test=".j:EmploymentLocation/j:LocationAddress/j:AddressFullText">
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select=".j:EmploymentLocation/j:LocationAddress/j:AddressFullText"/>
                <xsl:with-param name="StartPos" select="$posCol2 - 8"/>
                <xsl:with-param name="EndPos" select="$posMaxWidth"/>
                <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
                <xsl:with-param name="CurrentPos" select="$posMaxWidth +
1"/>
                <xsl:with-param name="WrapWidth" select="$posMaxWidth +
1"/>
                <xsl:with-param name="initialRun" select="0"/>
            </xsl:call-template>
            <xsl:text>#10;</xsl:text>
        </xsl:if>
        <xsl:if test=".j:EmploymentLocation/j:LocationAddress/j:BuildingName">
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select=".j:EmploymentLocation/j:LocationAddress/j:BuildingName"/>
                <xsl:with-param name="StartPos" select="$posCol2 - 17"/>
                <xsl:with-param name="EndPos" select="$posMaxWidth"/>
                <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
                <xsl:with-param name="CurrentPos" select="$posMaxWidth +
1"/>
                <xsl:with-param name="WrapWidth" select="$posMaxWidth +
1"/>

```



```

                <xsl:with-param name="initialRun" select="0"/>
            </xsl:call-template>
        </xsl:if>
        <xsl:text>#10;</xsl:text>
        <xsl:if
test="./j:EmploymentLocation/j:LocationAddress/j:LocationStreet/j:StreetFullText">
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select="./j:EmploymentLocation/j:LocationAddress/j:StreetFullText"/>
                <xsl:with-param name="StartPos" select="$posCol2"/>
                <xsl:with-param name="EndPos" select="$posMaxWidth"/>
                <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
                <xsl:with-param name="CurrentPos" select="$posMaxWidth +
1"/>
                <xsl:with-param name="WrapWidth" select="$posMaxWidth +
1"/>
                <xsl:with-param name="initialRun" select="0"/>
            </xsl:call-template>
            <xsl:text>#10;</xsl:text>
        </xsl:if>
        <xsl:if
test="./j:EmploymentLocation/j:LocationAddress/j:LocationStreet/j:StreetNumberText">
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select="concat(./j:EmploymentLocation/j:LocationAddress/j:LocationStreet/j:StreetNumberText,'
./j:EmploymentLocation/j:LocationAddress/j:LocationStreet/j:StreetPredirectionalText,'
./j:EmploymentLocation/j:LocationAddress/j:LocationStreet/j:StreetName,'
./j:EmploymentLocation/j:LocationAddress/j:LocationStreet/j:StreetTypeText)"/>
                <xsl:with-param name="StartPos" select="$posCol2"/>
                <xsl:with-param name="EndPos" select="$posMaxWidth"/>
                <xsl:with-param name="WrappedStartPos" select="$posCol2"/>

```



1"/>

1"/>

```
<xsl:with-param name="StartPos" select="$posCol2"/>
<xsl:with-param name="EndPos" select="$posMaxWidth"/>
<xsl:with-param name="WrapedStartPos" select="$posCol2"/>
<xsl:with-param name="CurrentPos" select="$posMaxWidth +

<xsl:with-param name="WrapWidth" select="$posMaxWidth +

<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
</xsl:if>
</xsl:for-each>
<!-- Residence -->
<!-- Edited 05/03 -->
<xsl:choose>
  <xsl:when test="count(//rap:RapSheet/j:Subject/j:Residence) > 0">
    <xsl:text>Residence#10;</xsl:text>
  </xsl:when>
  <xsl:otherwise/>
</xsl:choose>
<!-- -->
<xsl:for-each select="//rap:RapSheet/j:Subject/j:Residence">
  <xsl:text>Residence as of </xsl:text>
  <xsl:if test="count(@j:reportedDate) > 0">
    <!-- If there is a reference date print it -->
    <xsl:value-of select="@j:reportedDate"/>
  </xsl:if>
  <xsl:text>#10;</xsl:text>
  <xsl:if test=". /j:LocationAddress/j:BuildingName">
    <xsl:call-template name="wrapIn">
```

```

select=".j:LocationAddress/j:BuildingName"/>
    <xsl:with-param name="Text"
    <xsl:with-param name="StartPos" select="$posCol2 - 17"/>
    <xsl:with-param name="EndPos" select="$posMaxWidth"/>
    <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
    <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
    <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
    <xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:choose>
    <xsl:when test=".j:LocationAddress/j:AddressFullText">
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text"
            <xsl:with-param name="StartPos" select="$posCol2"/>
            <xsl:with-param name="EndPos" select="$posMaxWidth"/>
            <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
            <xsl:with-param name="CurrentPos" select="$posMaxWidth +
            <xsl:with-param name="WrapWidth" select="$posMaxWidth +
            <xsl:with-param name="initialRun" select="0"/>
        </xsl:call-template>
        <xsl:text>#10;</xsl:text>
    </xsl:when>
    <xsl:otherwise>
        <xsl:if test=".j:LocationAddress/j:LocationStreet/j:StreetFullText">
            <xsl:call-template name="wrapIn">

```

```

select="./j:LocationAddress/j:LocationStreet/j:StreetFullText"/>

select="$posMaxWidth"/>

select="$posCol2"/>

select="$posMaxWidth + 1"/>

select="$posMaxWidth + 1"/>

select="./j:LocationAddress/j:LocationSecondaryUnitText"/>

select="$posMaxWidth"/>

select="$posCol2"/>

select="$posMaxWidth + 1"/>

select="$posMaxWidth + 1"/>

<xsl:with-param name="Text"

<xsl:with-param name="StartPos" select="$posCol2"/>
<xsl:with-param name="EndPos"

<xsl:with-param name="WrapedStartPos"

<xsl:with-param name="CurrentPos"

<xsl:with-param name="WrapWidth"

<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="./j:LocationAddress/j:LocationSecondaryUnitText">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"

<xsl:with-param name="StartPos" select="$posCol2"/>
<xsl:with-param name="EndPos"

<xsl:with-param name="WrapedStartPos"

<xsl:with-param name="CurrentPos"

<xsl:with-param name="WrapWidth"

<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>

```

```

select=".j:LocationAddress/j:AddressMailDeliveryUnitText"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
</xsl:if>
<xsl:if test=".j:LocationAddress/j:AddressMailDeliveryUnitText">
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"
      <xsl:with-param name="StartPos" select="$posCol2"/>
      <xsl:with-param name="EndPos"
        <xsl:with-param name="WrappedStartPos"
          <xsl:with-param name="CurrentPos"
            <xsl:with-param name="WrapWidth"
              <xsl:with-param name="initialRun" select="0"/>
            </xsl:call-template>
            <xsl:text>&#10;</xsl:text>
          </xsl:if>
          <xsl:if test=".j:LocationAddress/j:LocationStateName">
            <xsl:call-template name="wrapIn">
              <xsl:with-param name="Text"
                <xsl:with-param name="StartPos" select="$posCol2"/>
                <xsl:with-param name="EndPos"
                  <xsl:with-param name="WrappedStartPos"
                    <xsl:with-param name="CurrentPos"

```

```

select="$posMaxWidth + 1"/>
',./j:LocationAddress/j:LocationCountyName"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="./j:LocationAddress/j:LocationCountyName">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text" select="concat('County:
<xsl:with-param name="StartPos" select="$posCol2"/>
<xsl:with-param name="EndPos"
<xsl:with-param name="WrappedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
</xsl:otherwise>
</xsl:choose>
<xsl:choose>
<xsl:when test="count(./j:LocationContactInformation) > 0">
<!-- if there is a telephone number, write it -->
<xsl:for-each select="./j:LocationContactInformation">
<xsl:text>Telephone</xsl:text>
<xsl:choose>

```

```

test="count(/j:ContactTelephoneNumber/j:TelephoneNumberFullID) > 0"
-->
select="/j:ContactTelephoneNumber/j:TelephoneNumberFullID"/>
select="$posCol2 - 9"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
select="0"/>
<xsl:when
  <!-- if standard phone format, format it accordingly
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"
    <xsl:with-param name="StartPos"
    <xsl:with-param name="EndPos"
    <xsl:with-param name="WrappedStartPos"
    <xsl:with-param name="CurrentPos"
    <xsl:with-param name="WrapWidth"
    <xsl:with-param name="initialRun"
    <!-- 0=True -->
  </xsl:call-template>
</xsl:when>
<xsl:otherwise/>
</xsl:choose>
<xsl:text>#10;#10;</xsl:text>
</xsl:for-each>
</xsl:when>
<xsl:otherwise>
  <xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>

```



```

</xsl:for-each>
<!-- Fingerprint Images -->
<xsl:if test="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet">
  <xsl:text>Fingerprint Images&#10;</xsl:text>
</xsl:if>
<xsl:for-each
select="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonFingerprintSet/j:Fingerprint">
  <xsl:if test="j:BiometricImage/j:BinaryCaptureOrganization">
    <xsl:text>Fingerprint Image Available </xsl:text>
    <xsl:call-template name="wrapIn">
      <xsl:with-param name="Text"
select="concat(j:BiometricImage/j:BinaryCaptureOrganization/j:OrganizationName, ' ',
j:BiometricImage/j:BinaryCaptureOrganization/j:OrganizationORIID/j:ID)"/>
      <xsl:with-param name="StartPos" select="$amtColSpacer"/>
      <xsl:with-param name="EndPos" select="$posMaxWidth"/>
      <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
      <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
      <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
      <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>&#10;</xsl:text>
  </xsl:if>
  <xsl:if test="j:FingerprintFingerText">
    <xsl:text>Available Image </xsl:text>
    <xsl:call-template name="wrapIn">
      <xsl:with-param name="Text" select="j:FingerprintFingerText"/>
      <xsl:with-param name="StartPos" select="$amtColSpacer"/>
      <xsl:with-param name="EndPos" select="$posMaxWidth"/>
      <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
      <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
      <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
    </xsl:call-template>
  </xsl:if>
</xsl:for-each>

```

```

        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="j:BiometricImage/j:BinaryCaptureDate">
    <xsl:text>Capture Date    </xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"
select="j:BiometricImage/j:BinaryCaptureDate"/>
        <xsl:with-param name="StartPos" select="$amtColSpacer"/>
        <xsl:with-param name="EndPos" select="$posMaxWidth"/>
        <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="j:BiometricImage/j:BinaryReferenceID/j:ID">
    <xsl:text>Download URL    </xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"
select="j:BiometricImage/j:BinaryReferenceID/j:ID"/>
        <xsl:with-param name="StartPos" select="$amtColSpacer"/>
        <xsl:with-param name="EndPos" select="$posMaxWidth"/>
        <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>

```

```

</xsl:if>
<xsl:choose>
  <xsl:when test="j:BiometricImage/j:BinaryObject.Base64">
    <xsl:text>(Transmitted Image Suppressed: &#10;</xsl:text>
    <xsl:call-template name="wrapIn">
      <xsl:with-param name="Text"
select="concat('Type:'j:BiometricImage/j:BinaryTypeText, ' Format:'j:BiometricImage/j:BinaryFormatText, '
Size:'j:BiometricImage/j:BinarySizeValue,'K')"/>
      <xsl:with-param name="StartPos" select="$amtColSpacer"/>
      <xsl:with-param name="EndPos" select="$posMaxWidth"/>
      <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
      <xsl:with-param name="CurrentPos" select="$posMaxWidth +
1"/>
      <xsl:with-param name="WrapWidth" select="$posMaxWidth +
1"/>
      <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:call-template name="wrapIn">
      <xsl:with-param name="Text"
select="concat('Comment:'j:BiometricImage/j:BinaryDescriptionText,')"/>
      <xsl:with-param name="StartPos" select="$amtColSpacer"/>
      <xsl:with-param name="EndPos" select="$posMaxWidth"/>
      <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
      <xsl:with-param name="CurrentPos" select="$posMaxWidth +
1"/>
      <xsl:with-param name="WrapWidth" select="$posMaxWidth +
1"/>
      <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>&#10;</xsl:text>
  </xsl:when>

```

```

        <xsl:otherwise>
            <xsl:text>(No Fingerprint Image Transmitted </xsl:text>
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select="concat('&#10;Comment:',j:BiometricImage/j:BinaryDescriptionText)"/>
                <xsl:with-param name="StartPos" select="$amtColSpacer"/>
                <xsl:with-param name="EndPos" select="$posMaxWidth"/>
                <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
                <xsl:with-param name="CurrentPos" select="$posMaxWidth +
1"/>
                <xsl:with-param name="WrapWidth" select="$posMaxWidth +
1"/>
                <xsl:with-param name="initialRun" select="0"/>
            </xsl:call-template>
            <xsl:text>)&#10;</xsl:text>
        </xsl:otherwise>
    </xsl:choose>
</xsl:for-each>
<!-- Palmprint Images -->
<xsl:if test="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonPalmPrint">
    <xsl:text>&#10;Palmprint Images&#10;</xsl:text>
</xsl:if>
<xsl:for-each select="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonPalmPrint">
    <xsl:if test="j:BiometricImage/j:BinaryCaptureOrganization">
        <xsl:text>Palmprint Image Available </xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text"
select="concat(j:BiometricImage/j:BinaryCaptureOrganization/j:OrganizationName, ',
j:BiometricImage/j:BinaryCaptureOrganization/j:OrganizationORIID/j:ID)"/>
            <xsl:with-param name="StartPos" select="$amtColSpacer"/>
            <xsl:with-param name="EndPos" select="$posMaxWidth"/>

```

```

        <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="j:BiometricImage/j:BinaryTypeText">
    <xsl:text>Available Image </xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"
select="j:BiometricImage/j:BinaryTypeText"/>
        <xsl:with-param name="StartPos" select="$amtColSpacer"/>
        <xsl:with-param name="EndPos" select="$posMaxWidth"/>
        <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="j:BiometricImage/j:BinaryCaptureDate">
    <xsl:text>Capture Date </xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"
select="j:BiometricImage/j:BinaryCaptureDate"/>
        <xsl:with-param name="StartPos" select="$amtColSpacer"/>
        <xsl:with-param name="EndPos" select="$posMaxWidth"/>
        <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>

```

```

        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="j:BiometricImage/j:BinaryReferenceID/j:ID">
    <xsl:text>Download URL </xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"
select="j:BiometricImage/j:BinaryReferenceID/j:ID"/>
        <xsl:with-param name="StartPos" select="$amtColSpacer"/>
        <xsl:with-param name="EndPos" select="$posMaxWidth"/>
        <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:choose>
    <xsl:when test="j:BiometricImage/j:BinaryObject.Base64">
        <xsl:text>(Transmitted Image Suppressed: #10;</xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text"
select="concat('Type:'j:BiometricImage/j:BinaryTypeText, ' Format:'j:BiometricImage/j:BinaryFormatText, '
Size:'j:BiometricImage/j:BinarySizeValue,'K)"/>
            <xsl:with-param name="StartPos" select="$amtColSpacer"/>
            <xsl:with-param name="EndPos" select="$posMaxWidth"/>
            <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
            <xsl:with-param name="CurrentPos" select="$posMaxWidth +
1"/>

```

```

1"/>
<xsl:with-param name="WrapWidth" select="$posMaxWidth +
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:if test="j:BiometricImage/j:BinaryDescriptionText">
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"
select="concat('Comment:',j:BiometricImage/j:BinaryDescriptionText,')"/>
    <xsl:with-param name="StartPos"
select="$amtColSpacer"/>
    <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
    <xsl:with-param name="WrapedStartPos"
select="$posCol2"/>
    <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
    <xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>
    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
</xsl:if>
<xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
  <xsl:text>(No Palmprint Image Transmitted </xsl:text>
  <xsl:if test="j:BiometricImage/j:BinaryDescriptionText">
    <xsl:call-template name="wrapIn">
      <xsl:with-param name="Text"
select="concat('&#10;Comment:',j:BiometricImage/j:BinaryDescriptionText)"/>
      <xsl:with-param name="StartPos"
select="$amtColSpacer"/>

```

```

select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
</xsl:if>
<xsl:text>)&#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<!-- Photo Images -->
<xsl:if test="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonDigitalImage">
<xsl:text>&#10;Photo Images&#10;</xsl:text>
</xsl:if>
<xsl:for-each select="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonDigitalImage">
<xsl:if test="j:BinaryCaptureOrganization">
<xsl:text>Photo Image Available </xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
select="concat(j:BinaryCaptureOrganization/j:OrganizationName, ' ', j:BinaryCaptureOrganization/j:OrganizationORIID/j:ID)"/>
<xsl:with-param name="StartPos" select="$amtColSpacer"/>
<xsl:with-param name="EndPos" select="$posMaxWidth"/>
<xsl:with-param name="WrapedStartPos" select="$posCol2"/>
<xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
<xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
<xsl:with-param name="initialRun" select="0"/>

```



```

        </xsl:call-template>
        <xsl:text>&#10;</xsl:text>
    </xsl:if>
    <xsl:if test="j:BinaryTypeText">
        <xsl:text>Available Image </xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text" select="j:BinaryTypeText"/>
            <xsl:with-param name="StartPos" select="$amtColSpacer"/>
            <xsl:with-param name="EndPos" select="$posMaxWidth"/>
            <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
            <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
            <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
            <xsl:with-param name="initialRun" select="0"/>
        </xsl:call-template>
        <xsl:text>&#10;</xsl:text>
    </xsl:if>
    <xsl:if test="j:BinaryCaptureDate">
        <xsl:text>Capture Date </xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text" select="j:BinaryCaptureDate"/>
            <xsl:with-param name="StartPos" select="$amtColSpacer"/>
            <xsl:with-param name="EndPos" select="$posMaxWidth"/>
            <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
            <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
            <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
            <xsl:with-param name="initialRun" select="0"/>
        </xsl:call-template>
        <xsl:text>&#10;</xsl:text>
    </xsl:if>
    <xsl:if test="j:BinaryReferenceID/j:ID">
        <xsl:text>Download URL </xsl:text>
    </xsl:if>

```

```

        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text" select="j:BinaryReferenceID/j:ID"/>
            <xsl:with-param name="StartPos" select="$amtColSpacer"/>
            <xsl:with-param name="EndPos" select="$posMaxWidth"/>
            <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
            <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
            <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
            <xsl:with-param name="initialRun" select="0"/>
        </xsl:call-template>
        <xsl:text>#10;</xsl:text>
    </xsl:if>
    <xsl:choose>
        <xsl:when test="j:BinaryObject.Base64">
            <xsl:text>(Transmitted Image Suppressed: #10;</xsl:text>
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select="concat('Type:',j:BinaryTypeText, ' Format:',j:BinaryFormatText, ' Size:',j:BinarySizeValue,'K')"/>
                <xsl:with-param name="StartPos" select="$amtColSpacer"/>
                <xsl:with-param name="EndPos" select="$posMaxWidth"/>
                <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
                <xsl:with-param name="CurrentPos" select="$posMaxWidth +
1"/>
                <xsl:with-param name="WrapWidth" select="$posMaxWidth +
1"/>
                <xsl:with-param name="initialRun" select="0"/>
            </xsl:call-template>
            <xsl:if test="j:BinaryDescriptionText">
                <xsl:call-template name="wrapIn">
                    <xsl:with-param name="Text"
select="concat('Comment:',j:BinaryDescriptionText,')')"/>

```

```

select="$amtColSpacer"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
<xsl:with-param name="StartPos"
<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
</xsl:if>
<xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:text>(No Photo Image Transmitted </xsl:text>
<xsl:if test="j:BinaryDescriptionText">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
select="concat('&#10;Comment:'.j:BinaryDescriptionText)"/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

```

```

                <xsl:with-param name="initialRun" select="0"/>
            </xsl:call-template>
        </xsl:if>
        <xsl:text>)&#10;</xsl:text>
    </xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<!-- DNA -->
<!-- Edited 05/03 -->
<xsl:if test="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonDNA">
    <xsl:text>&#10;DNA Data&#10;</xsl:text>
</xsl:if>
<xsl:for-each select="//rap:RapSheet/j:Subject/j:PersonBiometricDetails/j:PersonDNA">
    <xsl:if test="j:BiometricCaptureDate">
        <xsl:text>DNA Sample Taken    </xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text" select="j:BiometricCaptureDate"/>
            <xsl:with-param name="StartPos" select="$amtColSpacer"/>
            <xsl:with-param name="EndPos" select="$posMaxWidth"/>
            <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
            <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
            <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
            <xsl:with-param name="initialRun" select="0"/>
        </xsl:call-template>
        <xsl:text>&#10;</xsl:text>
    </xsl:if>
    <xsl:if test="j:BiometricCaptureOrganization">
        <xsl:text>DNA Information Available </xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text"
select="concat(j:BinaryCaptureOrganization/j:OrganizationName, ' ', j:BinaryCaptureOrganization/j:OrganizationORIID/j:ID)"/>

```

```

        <xsl:with-param name="StartPos" select="$amtColSpacer"/>
        <xsl:with-param name="EndPos" select="$posMaxWidth"/>
        <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>&#10;</xsl:text>
</xsl:if>
<xsl:choose>
    <xsl:when test="j:DNALocus">
        <xsl:text>(Transmitted DNA Detail Suppressed</xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text" select="concat('; Encoding
Method',j:BiometricEncodingMethodText)"/>
            <xsl:with-param name="StartPos" select="$amtColSpacer"/>
            <xsl:with-param name="EndPos" select="$posMaxWidth"/>
            <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
            <xsl:with-param name="CurrentPos" select="$posMaxWidth +
1"/>
            <xsl:with-param name="WrapWidth" select="$posMaxWidth +
1"/>
            <xsl:with-param name="initialRun" select="0"/>
        </xsl:call-template>
        </xsl:when>
        <xsl:otherwise>
            <xsl:text>(No DNA Detail Transmitted </xsl:text>
        </xsl:otherwise>
    </xsl:choose>
<xsl:if test="j:BiometricDescriptionText">
    <xsl:call-template name="wrapIn">

```

```

        <xsl:with-param name="Text"
select="concat('&#10;Comment:',j:BiometricDescriptionText)"/>
        <xsl:with-param name="StartPos" select="$amtColSpacer"/>
        <xsl:with-param name="EndPos" select="$posMaxWidth"/>
        <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
</xsl:if>
<xsl:text>)&#10;</xsl:text>
</xsl:for-each>

<!-- Caution Information -->
<xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonFirearmSalesDisqualifiedIndicator) &gt; 0
or count(//rap:RapSheet/j:Subject/j:SubjectCautionInformationCaveat) &gt; 0 or
count(//rap:RapSheet/j:Subject/j:SubjectOffenderNoticeCaveat) &gt; 0">
    <xsl:text>&#10;Caution Information&#10;</xsl:text>
</xsl:if>
<!-- Firearms Disqualified -->
<xsl:if test="count(//rap:RapSheet/j:Subject/j:PersonFirearmSalesDisqualifiedIndicator) &gt;
0">
    <xsl:text>Firearms Disqualified</xsl:text>
    <xsl:call-template name="spaceover">
        <xsl:with-param name="amount" select="$posCol2 - 21"/>
        <xsl:with-param name="amountWrote" select="0"/>
    </xsl:call-template>
    <xsl:value-of
select="//rap:RapSheet/j:Subject/j:PersonFirearmSalesDisqualifiedIndicator"/>
</xsl:if>

```

```

<!-- Caution -->
<xsl:for-each select="//rap:RapSheet/j:Subject/j:SubjectCautionInformationCaveat">
  <xsl:text>&#10;Caution</xsl:text>
  <xsl:variable name="CautionInfo">
    <xsl:value-of select="."/>
    <xsl:if test="./@j:reportedDate or ./@j:reportingOrganizationText">
      <xsl:text> (</xsl:text>
      <xsl:if test="./@j:reportingOrganizationText">
        <xsl:value-of select="./@j:reportingOrganizationText"/>
      </xsl:if>
      <xsl:if test="./@j:reportedDate and ./@j:reportingOrganizationText">
        <xsl:text>; </xsl:text>
      </xsl:if>
      <xsl:value-of select="./@j:reportedDate"/>
      <xsl:text>)</xsl:text>
    </xsl:if>
  </xsl:variable>
  <!-- djr(19-mar-03) Write out the Description Info now that it is all formatted. -->
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text" select="$CautionInfo"/>
    <xsl:with-param name="StartPos" select="$posCol2 - 7"/>
    <xsl:with-param name="EndPos" select="$posMaxWidth"/>
    <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
    <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
    <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
</xsl:for-each>
<!-- xsl:text>&#10;</xsl:text -->
<!-- Notice -->
<xsl:for-each select="//rap:RapSheet/j:Subject/j:SubjectOffenderNoticeCaveat">

```

```

<xsl:text>&#10;Notice</xsl:text>
<xsl:variable name="NoticeInfo">
  <xsl:value-of select="."/>
  <xsl:if test="./@j:reportedDate or ./@j:reportingOrganizationText">
    <xsl:text> (</xsl:text>
    <xsl:if test="./@j:reportingOrganizationText">
      <xsl:value-of select="./@j:reportingOrganizationText"/>
    </xsl:if>
    <xsl:if test="./@j:reportedDate and ./@j:reportingOrganizationText">
      <xsl:text>; </xsl:text>
    </xsl:if>
    <xsl:value-of select="./@j:reportedDate"/>
    <xsl:text>)</xsl:text>
  </xsl:if>
</xsl:variable>
<xsl:call-template name="wrapIn">
  <xsl:with-param name="Text" select="$NoticeInfo"/>
  <xsl:with-param name="StartPos" select="$posCol2 - (6)"/>
  <xsl:with-param name="EndPos" select="$posMaxWidth"/>
  <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
  <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
  <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
  <xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
</xsl:for-each>
<xsl:text>&#10;</xsl:text>

<!-- ##### Criminal History Section ##### -->
<xsl:text>&#10;***** CRIMINAL HISTORY
*****&#10;&#10;</xsl:text>
<xsl:for-each select="//rap:RapSheet/rap:Cycle">

```



```

<xsl:sort select="/rap:CycleEarliestDate"/>
<!-- dr: 17-OCT-02 sort all Cycles by earliestDate -->
<xsl:variable name="Cycle-number" select="position()"/>
<xsl:text>==== Cycle </xsl:text>
<xsl:value-of select="format-number($Cycle-number,'#000')"/>
<xsl:text>====&#10;</xsl:text>
<!-- Edited 05/03 -->
<xsl:if test="/rap:CycleTrackingNumber">
  <xsl:text>Tracking Number</xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text" select="/rap:CycleTrackingNumber"/>
    <xsl:with-param name="StartPos" select="$posCol2 - (15)"/>
    <xsl:with-param name="EndPos" select="$posMaxWidth"/>
    <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
    <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
    <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
  <xsl:text>&#10;</xsl:text>
</xsl:if>
<!-- -->
<!-- Edited 05/03 -->
<xsl:if test="/rap:CycleEarliestDate">
  <xsl:text>Earliest Event Date</xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text" select="/rap:CycleEarliestDate"/>
    <xsl:with-param name="StartPos" select="$posCol2 - (19)"/>
    <xsl:with-param name="EndPos" select="$posMaxWidth"/>
    <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
    <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
    <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
  </xsl:call-template>
</xsl:if>

```

```

        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
</xsl:if>
<xsl:if test="/j:Incident/j:ActivityDate">
    <xsl:text>Incident Date</xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text" select="/j:Incident/j:ActivityDate"/>
        <xsl:with-param name="StartPos" select="$posCol2 - (13)"/>
        <xsl:with-param name="EndPos" select="$posMaxWidth"/>
        <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
        <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
</xsl:if>
<!-- -->
<!-- Arrest Section -->
<xsl:choose>
    <!-- If there is any arrests print them -->
    <xsl:when test="count(j:Arrest) > 0">
        <!-- Print Arrest Info -->
        <xsl:text>&#10;-----
--&#10;</xsl:text>

        <!-- Edited 05/03 -->
        <xsl:if test="/j:Arrest/j:ActivityDate">
            <xsl:text>Arrest Date</xsl:text>
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select="/j:Arrest/j:ActivityDate"/>
                <xsl:with-param name="StartPos" select="$posCol2 -
(11)"/>

```

```

select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<!-- -->
<!-- Edited 05/03 -->
<xsl:if test=".j:Arrest/j:ArrestAgencyRecordID">
<xsl:text>Arrest Case Number</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos" select="$posCol2 -
<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>

```

```

        <xsl:text>&#10;</xsl:text>
    </xsl:if>
    <!-- -->
    <!-- Edited 05/03 -->
    <xsl:if test="./j:Arrest/j:ArrestAgency">
        <xsl:text>Arresting Agency</xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text"
select="concat('./j:Arrest/j:ArrestAgency/j:OrganizationORIID/j:ID,' ',./j:Arrest/j:ArrestAgency/j:OrganizationName)"/>
            <xsl:with-param name="StartPos" select="$posCol2 -
(16)"/>
            <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
            <xsl:with-param name="WrappedStartPos"
select="$posCol2"/>
            <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
            <xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>
            <xsl:with-param name="initialRun" select="0"/>
        </xsl:call-template>
        <xsl:text>&#10;</xsl:text>
    </xsl:if>
    <!-- -->
    <!-- Edited 05/03 -->
    <xsl:if test="./j:Arrest/j:ArrestSubject">
        <xsl:text>Subject's Name</xsl:text>
    </xsl:if>
    <!-- -->
    <xsl:for-each select="./j:Arrest/j:ArrestSubject/j:PersonName">
        <!-- Name -->

```

data -->

select="\$posCol2 - 14"/>

select="0"/>

select="\$posCol2"/>

select="0"/>

```
<xsl:variable name="i" select="position()"/>
<xsl:choose>
  <!-- determine the correct spacing before the write of name
  <xsl:when test="$i = 1">
    <xsl:call-template name="spaceover">
      <xsl:with-param name="amount"
      <xsl:with-param name="amountWrote"
    </xsl:call-template>
  </xsl:when>
  <xsl:otherwise>
    <xsl:call-template name="spaceover">
      <xsl:with-param name="amount"
      <xsl:with-param name="amountWrote"
    </xsl:call-template>
  </xsl:otherwise>
</xsl:choose>
<xsl:value-of select="./j:PersonFullName"/>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
<!--
<xsl:if test="not(./j:Arrest/j:PersonName)">
  <xsl:text>#10;</xsl:text>
</xsl:if>
-->
<!-- Removed Dashes 09/07 -->
<xsl:if test="./j:Arrest/j:ArrestSubject/j:SubjectID">
```

```
select=".j:Arrest/j:ArrestSubject/j:SubjectID/j:ID"/>
```

```
(22)"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select=".j:Arrest/j:ActivityTypeText"/>
```

```
(22)"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
<xsl:text> Offender Id Number</xsl:text>
```

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos" select="$posCol2 -
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun" select="0"/>
```

```
</xsl:call-template>
```

```
<xsl:text>&#10;</xsl:text>
```

```
</xsl:if>
```

```
<xsl:if test=".j:Arrest/j:ActivityTypeText">
```

```
<xsl:text>Arrest Type </xsl:text>
```

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos" select="$posCol2 -
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
select="$posMaxWidth + 1"/>
```

```
space."/>
```

```
(22)"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
<xsl:with-param name="WrapWidth"  
  <xsl:with-param name="initialRun" select="0"/>  
</xsl:call-template>  
<xsl:text>#10;</xsl:text>  
</xsl:if>  
<xsl:for-each select=".j:Arrest/j:ActivityCommentText">  
  <xsl:text>Comment(s) </xsl:text>  
  <xsl:call-template name="wrapIn">  
    <xsl:with-param name="Text" select="normalize-  
      <xsl:with-param name="StartPos" select="$posCol2 -  
        <xsl:with-param name="EndPos"  
          <xsl:with-param name="WrappedStartPos"  
            <xsl:with-param name="CurrentPos"  
              <xsl:with-param name="WrapWidth"  
                <xsl:with-param name="initialRun" select="0"/>  
              </xsl:call-template>  
            <xsl:text>#10;</xsl:text>  
          </xsl:for-each>  
        <xsl:for-each select=".j:Arrest/j:ArrestCharge">  
          <!-- Loop through the Arrest Charges -->  
          <!-- Removed Dashes 09/07 -->  
          <!--<xsl:text>Charge#10;</xsl:text -->  
          <xsl:if test="j:ChargeSequenceID/j:ID">  
            <xsl:text>Charge </xsl:text>
```

```

        <xsl:value-of select="j:ChargeSequenceID/j:ID"/>
        <xsl:text>#10;</xsl:text>
    </xsl:if>
    <xsl:if test="j:ChargeID/j:ID">
        <xsl:text>    Charge Number </xsl:text>
        <xsl:value-of select="j:ChargeID/j:ID"/>
        <xsl:text>#10;</xsl:text>
    </xsl:if>
    <xsl:if test="j:ChargeTrackingID">
        <xsl:text>Charge Tracking Number </xsl:text>
        <xsl:value-of select="j:ChargeTrackingID/j:ID"/>
        <xsl:text>#10;</xsl:text>
    </xsl:if>
    <xsl:if test="j:ChargeText">
        <xsl:text>    Charge Literal </xsl:text>
        <xsl:value-of select="j:ChargeText"/>
        <xsl:text>#10;</xsl:text>
    </xsl:if>
    <xsl:if test="./j:ArrestAgency/j:OrganizationORIID/j:ID or
./j:ArrestAgency/j:OrganizationName">
        <xsl:text>        Agency</xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text"
select="concat(./j:ArrestAgency/j:OrganizationORIID/j:ID, ' ',./j:ArrestAgency/j:OrganizationName)"/>
            <xsl:with-param name="StartPos"
select="$amtColSpacer"/>
            <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
            <xsl:with-param name="WrappedStartPos"
select="$posCol2"/>

```



```

select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
select="j:ChargeDescriptionText"/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="j:ChargeDescriptionText">
<xsl:text> Charge Description</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<xsl:with-param name="EndPos"
<xsl:with-param name="WrappedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="./j:ChargeStatute">
<xsl:text> Statute</xsl:text>
</xsl:if>
<xsl:for-each select="./j:ChargeStatute">

```

```

select="concat(/j:StatuteText, ' (, ./j:StatuteCodeID/j:ID, '; ', ./j:StatuteJurisdiction/j:LocationStateName, ')')"/>
select="$amtColSpacer"/>

select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

select="."/>
select="$posCol2"/>
-->

```

```

<xsl:variable name="i" select="position()"/>
<xsl:choose>
  <!-- determine the correct spacing and write it -->
  <xsl:when test="$i = 1">
    <xsl:call-template name="wrapIn">
      <xsl:with-param name="Text"
select="concat(/j:StatuteText, ' (, ./j:StatuteCodeID/j:ID, '; ', ./j:StatuteJurisdiction/j:LocationStateName, ')')"/>
      <xsl:with-param name="StartPos"
select="$amtColSpacer"/>
      <!-- space over spacer width -->
      <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
      <xsl:with-param
name="WrapedStartPos" select="$posCol2"/>
      <xsl:with-param
select="$posMaxWidth + 1"/>
      <xsl:with-param
name="WrapWidth" select="$posMaxWidth + 1"/>
      <xsl:with-param
select="0"/>
      </xsl:call-template>
      <xsl:text>&#10;</xsl:text>
    </xsl:when>
    <xsl:otherwise>
      <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"
select="."/>
        <xsl:with-param name="StartPos"
select="$posCol2"/>
        <!-- space over to Column2 start pos
-->

```

```

select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test="./j:ChargeReporting/j:ChargeNCICCode">
<xsl:text> NCIC Offense Code </xsl:text>
</xsl:if>
<xsl:for-each select="./j:ChargeReporting/j:ChargeNCICCode">
<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->
<xsl:when test="$i = 1">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over spacer width -->
<xsl:with-param name="EndPos"
select="."/>
select="$amtColSpacer - 3"/>
select="$posMaxWidth"/>

```

```

name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

select="."/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>

```

```

        </xsl:otherwise>
    </xsl:choose>
</xsl:for-each>
<xsl:if test="./j:ChargeReporting/j:ChargeNCICCode &lt; 1">
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="./j:ChargeStatute/j:StatuteOffenseID/j:ID">
    <xsl:text> State Offense Code</xsl:text>
</xsl:if>
<xsl:for-each select="./j:ChargeStatute/j:StatuteOffenseID/j:ID">
    <xsl:variable name="i" select="position()"/>
    <xsl:choose>
        <!-- determine the correct spacing and write it -->
        <xsl:when test="$i = 1">
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
                    <xsl:with-param name="StartPos"
                        <!-- space over spacer width -->
                        <xsl:with-param name="EndPos"
                            <xsl:with-param
                                <xsl:with-param name="CurrentPos"
                                    <xsl:with-param
                                        <xsl:with-param name="initialRun"
                                            </xsl:call-template>

```

```

select="."/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

        <xsl:text>&#10;</xsl:text>
    </xsl:when>
    <xsl:otherwise>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text"

            <xsl:with-param name="StartPos"

            <!-- space over to Column2 start pos

            <xsl:with-param name="EndPos"

            <xsl:with-param

            <xsl:with-param name="CurrentPos"

            <xsl:with-param

            <xsl:with-param name="initialRun"

        </xsl:call-template>
        <xsl:text>&#10;</xsl:text>
    </xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test="./j:ChargeStatute/j:StatuteOffenseID &lt; 1">
    <xsl:text>&#10;</xsl:text>
</xsl:if>
<xsl:if test="./j:ChargeCountQuantity">
    <xsl:text>        Counts</xsl:text>
    <xsl:call-template name="wrapIn">

```

```
select=".j:ChargeCountQuantity"/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select=".j:ChargeClassification/j:ChargeSeverityText"/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos"
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun" select="0"/>
```

```
</xsl:call-template>
```

```
<xsl:text>#10;</xsl:text>
```

```
</xsl:if>
```

```
<xsl:if test=".j:ChargeClassification/j:ChargeSeverityText">
```

```
<xsl:text> Severity</xsl:text>
```

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos"
```

```
<!-- space over spacer width -->
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```

select="$posMaxWidth + 1"/>

select=".j:ChargeClassification/j:ChargeApplicabilityText"/>

select="$amtColSpacer"/>

select="$posMaxWidth"/>

select="$posCol2"/>

select="$posMaxWidth + 1"/>

select="$posMaxWidth + 1"/>

test=".j:ChargeClassification/j:ChargeSpecialAllegationText">

select=".j:ChargeClassification/j:ChargeSpecialAllegationText">
    <xsl:with-param name="WrapWidth"
    <xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test=".j:ChargeClassification/j:ChargeApplicabilityText">
    <xsl:text> Inchoate Charge</xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"
        <xsl:with-param name="StartPos"
        <xsl:with-param name="EndPos"
        <xsl:with-param name="WrappedStartPos"
        <xsl:with-param name="CurrentPos"
        <xsl:with-param name="WrapWidth"
        <xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if
    <xsl:text> Enhancing Factor</xsl:text>
</xsl:if>
<xsl:for-each

```



```
select="."/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
name="WrapedStartPos" select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
name="WrapWidth" select="$posMaxWidth + 1"/>
```

```
select="0"/>
```

```
select="."/>
```

```
select="$posCol2"/>
```

```
-->
```

```
<xsl:variable name="i" select="position()"/>
```

```
<xsl:choose>
```

```
  <!-- determine the correct spacing and write it -->
```

```
  <xsl:when test="$i = 1">
```

```
    <xsl:call-template name="wrapIn">
```

```
      <xsl:with-param name="Text"
```

```
        <xsl:with-param name="StartPos"
```

```
          <!-- space over spacer width -->
```

```
          <xsl:with-param name="EndPos"
```

```
            <xsl:with-param
```

```
              <xsl:with-param name="CurrentPos"
```

```
                <xsl:with-param
```

```
                  <xsl:with-param name="initialRun"
```

```
                    </xsl:call-template>
```

```
                    <xsl:text>&#10;</xsl:text>
```

```
  </xsl:when>
```

```
  <xsl:otherwise>
```

```
    <xsl:call-template name="wrapIn">
```

```
      <xsl:with-param name="Text"
```

```
        <xsl:with-param name="StartPos"
```

```
          <!-- space over to Column2 start pos
```

```

select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

test=".j:ChargeClassification/j:ChargeReducingFactorText">
select=".j:ChargeClassification/j:ChargeReducingFactorText">

select="."/>
select="$amtColSpacer"/>

<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"

</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if
<xsl:text> Reducing Factor</xsl:text>
</xsl:if>
<xsl:for-each
<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->
<xsl:when test="$i = 1">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"

<xsl:with-param name="StartPos"

<!-- space over spacer width -->

```

```

select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

select="."/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"

```

```

        </xsl:call-template>
        <xsl:text>&#10;</xsl:text>
    </xsl:otherwise>
    </xsl:choose>
</xsl:for-each>
<xsl:if test="./j:ChargeDisposition">
    <xsl:text>        Disposition</xsl:text>
</xsl:if>
<xsl:for-each select="./j:ChargeDisposition">
    <xsl:variable name="i" select="position()"/>
    <xsl:choose>
        <!-- determine the correct spacing and write it -->
        <xsl:when test="$i = 1">
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select="concat('(./@j:commentText, './j:ChargeDispositionDate;', './j:ChargeDispositionDescriptionText, ')')"/>
                <xsl:with-param name="StartPos"
select="$amtColSpacer"/>
                <!-- space over spacer width -->
                <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
                <xsl:with-param
name="WrapedStartPos" select="$posCol2"/>
                <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
                <xsl:with-param
name="WrapWidth" select="$posMaxWidth + 1"/>
                <xsl:with-param name="initialRun"
select="0"/>
            </xsl:call-template>
            <xsl:text>&#10;</xsl:text>

```

```

select="concat(.,' ('./@j:reportedDate,',';','j:ChargeDispositionDescriptionText, ')')"/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"
    <xsl:with-param name="StartPos"
    <!-- space over to Column2 start pos
    <xsl:with-param name="EndPos"
    <xsl:with-param
    <xsl:with-param name="CurrentPos"
    <xsl:with-param
    <xsl:with-param name="initialRun"
  </xsl:call-template>
  <xsl:text>&#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test="j:ActivityCommentText">
  <xsl:text>      Comment</xsl:text>
</xsl:if>
<xsl:for-each select="j:ActivityCommentText">
  <xsl:variable name="i" select="position()"/>
  <xsl:choose>
    <!-- determine the correct spacing and write it -->

```

```

select="normalize-space(.)"/>
select="$amtColSpacer"/>

select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

select="."/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>

```

```

<xsl:when test="$i = 1">
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over spacer width -->
    <xsl:with-param name="EndPos"

    <xsl:with-param

    <xsl:with-param name="CurrentPos"

    <xsl:with-param

    <xsl:with-param name="initialRun"

  </xsl:call-template>
  <xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over to Column2 start pos

    <xsl:with-param name="EndPos"

```

```

name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
</xsl:for-each>
<!-- End of Charge Loop -->
</xsl:when>
<!-- End of Print of Arrest Info -->
<xsl:otherwise>
<!-- No Arrests to Print -->
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
<!-- Booking Section -->
<xsl:choose>
<!-- If there is any arrests print them -->
<xsl:when test="count(j:Booking) > 0">
<!-- Print Arrest Info -->
<xsl:text>#10;-----
--&#10;</xsl:text>
<!-- Edited 05/03 -->
<xsl:if test="./j:Booking/j:BookingAgencyRecordID">

```

```
select=".j:Booking/j:BookingAgencyRecordID/j:ID"/>
```

```
(19)">
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
<xsl:text>Booking Case Number</xsl:text>
```

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos" select="$posCol2 -
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun" select="0"/>
```

```
</xsl:call-template>
```

```
<xsl:text>&#10;</xsl:text>
```

```
</xsl:if>
```

```
<xsl:if test=".j:Booking/j:BookingAgency">
```

```
<xsl:text>Booking Agency</xsl:text>
```

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
select="concat(.j:Booking/j:BookingAgency/j:OrganizationName, ' ', j:Booking/j:BookingAgency/j:OrganizationORIID/j:ID)"/>
```

```
(14)">
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
<xsl:with-param name="StartPos" select="$posCol2 -
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```



```

select="$posMaxWidth + 1"/>
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
</xsl:when>
</xsl:choose>
<!-- Prosecutor Disposition Section -->
<xsl:choose>
<xsl:when test="count(rap:Prosecution) > 0">
<xsl:text>-----
#10;</xsl:text>
<xsl:text>Prosecutor Disposition (Cycle </xsl:text>
<xsl:value-of select="format-number($Cycle-number,'#000)"/>
<xsl:text>)</xsl:text>
<xsl:text>#10;</xsl:text>
<xsl:if test="/rap:Prosecution/j:ProsecutionCharge/j:ChargeID/j:ID">
<xsl:text>Prosecutor Case Number</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
select="/rap:Prosecution/j:ProsecutionCharge/j:ChargeID/j:ID"/>
22"/>
<xsl:with-param name="StartPos" select="$posCol2 -
<!-- space over to Column2 start pos -->
<xsl:with-param name="EndPos"
<xsl:with-param name="WrappedStartPos"
<xsl:with-param name="CurrentPos"
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>

```

```

select="$posMaxWidth + 1"/>
select="/rap:Prosecution/j:ActivityDate"/>
(16)/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
select="concat(/rap:Prosecution/rap:ProsecutionAgency/j:OrganizationORIID/j:ID,'
',./rap:Prosecution/rap:ProsecutionAgency/j:OrganizationName)"/>
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="/rap:Prosecution/j:ActivityDate">
<xsl:text>Prosecution Date</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos" select="$posCol2 -
<xsl:with-param name="EndPos"
<xsl:with-param name="WrappedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="/rap:Prosecution/rap:ProsecutionAgency">
<xsl:text>Prosecutor Agency</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"

```

```

17"/>

select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

<xsl:with-param name="StartPos" select="$posCol2 -
<!-- space over to Column2 start pos -->
<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:if>
<xsl:if test="/rap:Prosecution/rap:ProsecutionSubject/j:PersonName">
  <xsl:text>Subject's Name</xsl:text>
</xsl:if>
<xsl:for-each
select="/rap:Prosecution/rap:ProsecutionSubject/j:PersonName">
  <!-- Name -->
  <xsl:variable name="i" select="position()"/>
  <xsl:choose>
    <!-- determine the correct spacing before the write of name
  <xsl:when test="$i = 1">
    <xsl:call-template name="spaceover">
      <xsl:with-param name="amount"
    <xsl:with-param name="amountWrote"
select="$posCol2 - 14"/>
select="0"/>

```

```

select="$posCol2"/>
select="0"/>
</xsl:call-template>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="spaceover">
    <xsl:with-param name="amount"
      <xsl:with-param name="amountWrote"

```

```

</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
<xsl:value-of select="./j:PersonFullName"/>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
<xsl:for-each select="./rap:Prosecution/j:ProsecutionCharge">
  <!-- Loop through the Prosecutor Charges -->
  <xsl:if test="j:ChargeSequenceID/j:ID">
    <xsl:text>Charge </xsl:text>
    <xsl:value-of select="j:ChargeSequenceID/j:ID"/>
    <xsl:text>#10;</xsl:text>
  </xsl:if>
  <xsl:if test="j:ChargeID/j:ID">
    <xsl:text> Charge Number </xsl:text>
    <xsl:value-of select="j:ChargeID/j:ID"/>
    <xsl:text>#10;</xsl:text>
  </xsl:if>
  <xsl:if test="j:ChargeTrackingID">
    <xsl:text>Charge Tracking Number </xsl:text>
    <xsl:value-of select="j:ChargeTrackingID/j:ID"/>
    <xsl:text>#10;</xsl:text>
  </xsl:if>

```

```
select="j:ChargeDescriptionText"/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
<xsl:if test="j:ChargeText">
  <xsl:text> Charge Literal </xsl:text>
  <xsl:value-of select="j:ChargeText"/>
  <xsl:text>&#10;</xsl:text>
</xsl:if>
<xsl:if test="j:ChargeDescriptionText">
  <xsl:text> Charge Description</xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"
      <xsl:with-param name="StartPos"
      <xsl:with-param name="EndPos"
      <xsl:with-param name="WrappedStartPos"
      <xsl:with-param name="CurrentPos"
      <xsl:with-param name="WrapWidth"
      <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>&#10;</xsl:text>
  </xsl:if>
  <xsl:if test="./j:ChargeStatute">
    <xsl:text> Statute</xsl:text>
  </xsl:if>
  <xsl:for-each select="./j:ChargeStatute">
    <xsl:variable name="i" select="position()"/>
    <xsl:choose>
      <!-- determine the correct spacing and write it -->
```

```

select="concat(/j:StatuteText, ' (, /j:StatuteCodeID/j:ID, '; ', /j:StatuteJurisdiction/j:LocationStateName, ')')"/>
select="$amtColSpacer"/>

select="$posMaxWidth"/>

name="WrapedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

name="WrapWidth" select="$posMaxWidth + 1"/>

select="0"/>

select="."/>

select="$posCol2"/>

-->

select="$posMaxWidth"/>

```

```

<xsl:when test="$i = 1">
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"
    <xsl:with-param name="StartPos"

    <!-- space over spacer width -->
    <xsl:with-param name="EndPos"

    <xsl:with-param

    <xsl:with-param name="CurrentPos"

    <xsl:with-param

    <xsl:with-param name="initialRun"

  </xsl:call-template>
  <xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over to Column2 start pos

    <xsl:with-param name="EndPos"

```

```

name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test="./j:ChargeReporting/j:ChargeNCICCode">
<xsl:text> NCIC Offense Code </xsl:text>
</xsl:if>
<xsl:for-each select="./j:ChargeReporting/j:ChargeNCICCode">
<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->
<xsl:when test="$i = 1">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over spacer width -->
<xsl:with-param name="EndPos"
<xsl:with-param
select="."/>
select="$amtColSpacer - 3"/>
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>

```

```

select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

select="."/>

```

```

select="$posCol2"/>

```

```

-->

```

```

select="$posMaxWidth"/>

```

```

name="WrapedStartPos" select="$posCol2"/>

```

```

select="$posMaxWidth + 1"/>

```

```

name="WrapWidth" select="$posMaxWidth + 1"/>

```

```

select="0"/>

```

```

<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>

```



```

select="."/>

select="$amtColSpacer"/>

select="$posMaxWidth"/>

name="WrapedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

name="WrapWidth" select="$posMaxWidth + 1"/>

select="0"/>

</xsl:for-each>
<xsl:if test=".j:ChargeReporting/j:ChargeNCICCode &lt; 1">
  <xsl:text>&#10;</xsl:text>
</xsl:if>
<xsl:if test=".j:ChargeStatute/j:StatuteOffenseID/j:ID">
  <xsl:text> State Offense Code</xsl:text>
</xsl:if>
<xsl:for-each select=".j:ChargeStatute/j:StatuteOffenseID/j:ID">
  <xsl:variable name="i" select="position()"/>
  <xsl:choose>
    <!-- determine the correct spacing and write it -->
    <xsl:when test="$i = 1">
      <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"

        <xsl:with-param name="StartPos"

        <!-- space over spacer width -->
        <xsl:with-param name="EndPos"

        <xsl:with-param

        <xsl:with-param name="CurrentPos"

        <xsl:with-param

        <xsl:with-param name="initialRun"

      </xsl:call-template>
      <xsl:text>&#10;</xsl:text>
    </xsl:when>

```

```

select="."/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

select="./j:ChargeCountQuantity"/>

```

```

<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over to Column2 start pos

    <xsl:with-param name="EndPos"

    <xsl:with-param

    <xsl:with-param name="CurrentPos"

    <xsl:with-param

    <xsl:with-param name="initialRun"

    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
  </xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test="./j:ChargeStatute/j:StatueOffenseID &lt; 1">
  <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="./j:ChargeCountQuantity">
  <xsl:text>          Counts</xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="./j:ChargeClassification/j:ChargeSeverityText"/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
<xsl:with-param name="StartPos"
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun" select="0"/>
```

```
</xsl:call-template>
```

```
<xsl:text>&#10;</xsl:text>
```

```
</xsl:if>
```

```
<xsl:if test="./j:ChargeClassification/j:ChargeSeverityText">
```

```
<xsl:text> Severity</xsl:text>
```

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos"
```

```
<!-- space over spacer width -->
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun" select="0"/>
```

```

select=".j:ChargeClassification/j:ChargeApplicabilityText"/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

test=".j:ChargeClassification/j:ChargeSpecialAllegationText">
select=".j:ChargeClassification/j:ChargeSpecialAllegationText">

        </xsl:call-template>
        <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test=".j:ChargeClassification/j:ChargeApplicabilityText">
    <xsl:text> Inchoate Charge</xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"

        <xsl:with-param name="StartPos"

        <xsl:with-param name="EndPos"

        <xsl:with-param name="WrapedStartPos"

        <xsl:with-param name="CurrentPos"

        <xsl:with-param name="WrapWidth"

        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if

    <xsl:text> Enhancing Factor</xsl:text>
</xsl:if>
<xsl:for-each

    <xsl:variable name="i" select="position()"/>
    <xsl:choose>
        <!-- determine the correct spacing and write it -->

```

```

select="."/>

select="$amtColSpacer"/>

select="$posMaxWidth"/>

name="WrapedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

name="WrapWidth" select="$posMaxWidth + 1"/>

select="0"/>

select="."/>

select="$posCol2"/>

-->

select="$posMaxWidth"/>

```

```

<xsl:when test="$i = 1">
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over spacer width -->
    <xsl:with-param name="EndPos"

    <xsl:with-param

    <xsl:with-param name="CurrentPos"

    <xsl:with-param

    <xsl:with-param name="initialRun"

  </xsl:call-template>
  <xsl:text>#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over to Column2 start pos

    <xsl:with-param name="EndPos"

```

```

name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

test=".j:ChargeClassification/j:ChargeReducingFactorText">
select=".j:ChargeClassification/j:ChargeReducingFactorText">

select="."/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>

<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"

</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if
<xsl:text> Reducing Factor</xsl:text>
</xsl:if>
<xsl:for-each
<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->
<xsl:when test="$i = 1">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"

<xsl:with-param name="StartPos"

<!-- space over spacer width -->
<xsl:with-param name="EndPos"

```

```

name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

select="."/>
select="$posCol2"/>
-->

```

```

select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>

```

```

        </xsl:otherwise>
    </xsl:choose>
</xsl:for-each>
<xsl:if test="./j:ChargeDisposition">
    <xsl:text>    Disposition</xsl:text>
</xsl:if>
<xsl:for-each select="./j:ChargeDisposition">
    <xsl:variable name="i" select="position()"/>
    <xsl:choose>
        <!-- determine the correct spacing and write it -->
        <xsl:when test="$i = 1">
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select="concat(' ,./@j:commentText, ' ,j:ChargeDispositionDate, ' ,j:ChargeDispositionDescriptionText, ')" />
                <xsl:with-param name="StartPos"
select="$amtColSpacer" />
                <!-- space over spacer width -->
                <xsl:with-param name="EndPos"
select="$posMaxWidth" />
                <xsl:with-param
name="WrapedStartPos" select="$posCol2" />
                <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1" />
                <xsl:with-param
name="WrapWidth" select="$posMaxWidth + 1" />
                <xsl:with-param name="initialRun"
select="0" />
            </xsl:call-template>
            <xsl:text>&#10;</xsl:text>
        </xsl:when>
        <xsl:otherwise>

```



```

select="concat(.,' (./@j:reportedDate;', 'j:ChargeDispositionDescriptionText, ')')"/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

<xsl:call-template name="wrapIn">
  <xsl:with-param name="Text"
  <xsl:with-param name="StartPos"
  <!-- space over to Column2 start pos
  <xsl:with-param name="EndPos"
  <xsl:with-param
  <xsl:with-param name="CurrentPos"
  <xsl:with-param
  <xsl:with-param name="initialRun"

```

```

  </xsl:call-template>
  <xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test="j:ActivityCommentText">
  <xsl:text>      Comment</xsl:text>
</xsl:if>
<xsl:for-each select="j:ActivityCommentText">
  <xsl:variable name="i" select="position()"/>
  <xsl:choose>
    <!-- determine the correct spacing and write it -->
    <xsl:when test="$i = 1">
      <xsl:call-template name="wrapIn">

```

```

select="normalize-space(.)"/>

select="$amtColSpacer"/>

select="$posMaxWidth"/>

name="WrapedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

name="WrapWidth" select="$posMaxWidth + 1"/>

select="0"/>

select="."/>

select="$posCol2"/>

-->

select="$posMaxWidth"/>

name="WrapedStartPos" select="$posCol2"/>

```

```

<xsl:with-param name="Text"

<xsl:with-param name="StartPos"

<!-- space over spacer width -->
<xsl:with-param name="EndPos"

<xsl:with-param

<xsl:with-param name="CurrentPos"

<xsl:with-param

<xsl:with-param name="initialRun"

</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over to Column2 start pos

    <xsl:with-param name="EndPos"

    <xsl:with-param

```

```

select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
</xsl:for-each>
<!-- End of Charge Loop -->
</xsl:when>
</xsl:choose>
<!-- Court Disposition Section -->
<xsl:choose>
<xsl:when test="count(rap:CourtAction) > 0">
<xsl:text>-----
#10;</xsl:text>
<xsl:text>Court Disposition (Cycle </xsl:text>
<xsl:value-of select="format-number($Cycle-number,'#000')"/>
<xsl:text>)</xsl:text>
<xsl:text>#10;</xsl:text>
<xsl:if test="rap:CourtAction/rap:CourtRecordID/j:ID">
<xsl:text>Court Case Number</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
select="rap:CourtAction/rap:CourtRecordID/j:ID"/>
17"/>
<xsl:with-param name="StartPos" select="$posCol2 -

```

```

select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

<!-- space over to Column2 start pos -->
<xsl:with-param name="EndPos"

<xsl:with-param name="WrappedStartPos"

<xsl:with-param name="CurrentPos"

<xsl:with-param name="WrapWidth"

<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="/rap:CourtAction/j:Court">
<xsl:text>Court Agency</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
select="concat(/rap:CourtAction/j:Court/j:OrganizationORIID/j:ID,' ',/rap:CourtAction/rap:Court/j:OrganizationName)"/>
12"/>
<xsl:with-param name="StartPos" select="$posCol2 -

<!-- space over to Column2 start pos -->
<xsl:with-param name="EndPos"

<xsl:with-param name="WrappedStartPos"

<xsl:with-param name="CurrentPos"

<xsl:with-param name="WrapWidth"

<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>

```

```

        <xsl:text>&#10;</xsl:text>
    </xsl:if>
    <xsl:if test="./rap:CourtAction/rap:CourtSubject/j:PersonName">
        <xsl:text>Subject's Name</xsl:text>
    </xsl:if>
    <xsl:for-each select="./rap:CourtAction/rap:CourtSubject/j:PersonName">
        <!-- Name -->
        <xsl:variable name="i" select="position()"/>
        <xsl:choose>
            <!-- determine the correct spacing before the write of name
data -->
            <xsl:when test="$i = 1">
                <xsl:call-template name="spaceover">
                    <xsl:with-param name="amount"
select="$posCol2 - 14"/>
                    <xsl:with-param name="amountWrote"
select="0"/>
                </xsl:call-template>
            </xsl:when>
            <xsl:otherwise>
                <xsl:call-template name="spaceover">
                    <xsl:with-param name="amount"
select="$posCol2"/>
                    <xsl:with-param name="amountWrote"
select="0"/>
                </xsl:call-template>
            </xsl:otherwise>
        </xsl:choose>
        <xsl:value-of select="./j:PersonFullName"/>
        <xsl:text>&#10;</xsl:text>
    </xsl:for-each>

```

```

<xsl:for-each select="/rap:CourtAction/j:CourtCharge">
  <!-- Loop through the Court Charges -->
  <!-- Removed Dashes 09/07 -->
  <xsl:if test="j:ChargeSequenceID/j:ID">
    <xsl:text>Charge </xsl:text>
    <xsl:value-of select="j:ChargeSequenceID/j:ID"/>
    <xsl:text>#10;</xsl:text>
  </xsl:if>
  <xsl:if test="j:ChargeID/j:ID">
    <xsl:text> Charge Number </xsl:text>
    <xsl:value-of select="j:ChargeID/j:ID"/>
    <xsl:text>#10;</xsl:text>
  </xsl:if>
  <xsl:if test="j:ChargeTrackingID">
    <xsl:text>Charge Tracking Number </xsl:text>
    <xsl:value-of select="j:ChargeTrackingID/j:ID"/>
    <xsl:text>#10;</xsl:text>
  </xsl:if>
  <xsl:if test="j:ChargeText">
    <xsl:text> Charge Literal </xsl:text>
    <xsl:value-of select="j:ChargeText"/>
    <xsl:text>#10;</xsl:text>
  </xsl:if>
  <xsl:if test="j:ChargeDescriptionText">
    <xsl:text> Charge Description</xsl:text>
    <xsl:call-template name="wrapIn">
      <xsl:with-param name="Text"
        <xsl:with-param name="StartPos"

```

```
select="j:ChargeDescriptionText"/>
```

```
select="$amtColSpacer"/>
```

```

select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test=".j:ChargeStatute">
<xsl:text> Statute</xsl:text>
</xsl:if>
<xsl:for-each select=".j:ChargeStatute">
<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->
<xsl:when test="$i = 1">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over spacer width -->
<xsl:with-param name="EndPos"
<xsl:with-param
select="concat(.j:StatuteText, ' (, .j:StatuteCodeID/j:ID, ', ', .j:StatuteJurisdiction/j:LocationStateName, ')')"/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>

```

```

select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

select="."/>

```

```

select="$posCol2"/>

```

```

-->

```

```

select="$posMaxWidth"/>

```

```

name="WrapedStartPos" select="$posCol2"/>

```

```

select="$posMaxWidth + 1"/>

```

```

name="WrapWidth" select="$posMaxWidth + 1"/>

```

```

select="0"/>

```

```

<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>

```



```
select="."/>
```

```
select="$amtColSpacer - 3"/>
```

```
select="$posMaxWidth"/>
```

```
name="WrapedStartPos" select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
name="WrapWidth" select="$posMaxWidth + 1"/>
```

```
select="0"/>
```

```
</xsl:for-each>  
<xsl:if test=".j:ChargeReporting/j:ChargeNCICCode">  
  <xsl:text> NCIC Offense Code </xsl:text>  
</xsl:if>  
<xsl:for-each select=".j:ChargeReporting/j:ChargeNCICCode">  
  <xsl:variable name="i" select="position()"/>  
  <xsl:choose>  
    <!-- determine the correct spacing and write it -->  
    <xsl:when test="$i = 1">  
      <xsl:call-template name="wrapIn">  
        <xsl:with-param name="Text"  
  
          <xsl:with-param name="StartPos"  
  
            <!-- space over spacer width -->  
            <xsl:with-param name="EndPos"  
  
              <xsl:with-param  
  
                <xsl:with-param name="CurrentPos"  
  
                  <xsl:with-param  
  
                    <xsl:with-param name="initialRun"  
  
                      </xsl:call-template>  
                      <xsl:text>&#10;</xsl:text>  
                    </xsl:when>  
                    <xsl:otherwise>  
                      <xsl:call-template name="wrapIn">
```

```

select="."/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"

```

```

</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test="./j:ChargeReporting/j:ChargeNCICCode &lt; 1">
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="./j:ChargeStatute/j:StatuteOffenseID/j:ID">
<xsl:text> State Offense Code</xsl:text>
</xsl:if>
<xsl:for-each select="./j:ChargeStatute/j:StatuteOffenseID/j:ID">
<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->

```

```

select="."/>

select="$amtColSpacer"/>

select="$posMaxWidth"/>

name="WrapedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

name="WrapWidth" select="$posMaxWidth + 1"/>

select="0"/>

select="."/>

select="$posCol2"/>

-->

select="$posMaxWidth"/>

```

```

<xsl:when test="$i = 1">
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over spacer width -->
    <xsl:with-param name="EndPos"

    <xsl:with-param

    <xsl:with-param name="CurrentPos"

    <xsl:with-param

    <xsl:with-param name="initialRun"

  </xsl:call-template>
  <xsl:text>#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over to Column2 start pos

    <xsl:with-param name="EndPos"

```

```

name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"

</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test="./j:ChargeStatute/j:StatueOffenseID < 1">
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="./j:ChargeCountQuantity">
<xsl:text> Counts</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"

<xsl:with-param name="StartPos"
<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"

select="./j:ChargeCountQuantity"/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

```

```
select=".j:ChargeClassification/j:ChargeSeverityText"/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select=".j:ChargeClassification/j:ChargeApplicabilityText"/>
```

```
select="$amtColSpacer"/>
```

```
        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test=".j:ChargeClassification/j:ChargeSeverityText">
    <xsl:text>        Severity</xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"

        <xsl:with-param name="StartPos"

        <!-- space over spacer width -->
        <xsl:with-param name="EndPos"

        <xsl:with-param name="WrappedStartPos"

        <xsl:with-param name="CurrentPos"

        <xsl:with-param name="WrapWidth"

        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test=".j:ChargeClassification/j:ChargeApplicabilityText">
    <xsl:text>        Inchoate Charge</xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"

        <xsl:with-param name="StartPos"
```

```

select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

test="./j:ChargeClassification/j:ChargeSpecialAllegationText">
select="./j:ChargeClassification/j:ChargeSpecialAllegationText">

select="."/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>

<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if
<xsl:text> Enhancing Factor</xsl:text>
</xsl:if>
<xsl:for-each
<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->
<xsl:when test="$i = 1">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over spacer width -->
<xsl:with-param name="EndPos"

```

```

name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

select="."/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>

```

```

test=".j:ChargeClassification/j:ChargeReducingFactorText">

select=".j:ChargeClassification/j:ChargeReducingFactorText">

select="."/>

select="$amtColSpacer"/>

select="$posMaxWidth"/>

name="WraPedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

name="WrapWidth" select="$posMaxWidth + 1"/>

select="0"/>

```

```

</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if
    <xsl:text> Reducing Factor</xsl:text>
</xsl:if>
<xsl:for-each
    <xsl:variable name="i" select="position()"/>
    <xsl:choose>
        <!-- determine the correct spacing and write it -->
        <xsl:when test="$i = 1">
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
                    <xsl:with-param name="StartPos"
                        <!-- space over spacer width -->
                        <xsl:with-param name="EndPos"
                            <xsl:with-param
                                <xsl:with-param name="CurrentPos"
                                    <xsl:with-param
                                        <xsl:with-param name="initialRun"
                                            </xsl:call-template>
                                            <xsl:text>&#10;</xsl:text>

```



```

select="."/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

</xsl:when>
<xsl:otherwise>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"
            <xsl:with-param name="StartPos"
                <!-- space over to Column2 start pos
            <xsl:with-param name="EndPos"
                <xsl:with-param
                    <xsl:with-param name="CurrentPos"
                        <xsl:with-param
                            <xsl:with-param name="initialRun"
                                </xsl:call-template>
                                <xsl:text>&#10;</xsl:text>
                            </xsl:otherwise>
                        </xsl:choose>
                    </xsl:for-each>
                <xsl:if test=".j:ChargeDisposition">
                    <xsl:text> Disposition</xsl:text>
                </xsl:if>
            <xsl:for-each select=".j:ChargeDisposition">
                <xsl:variable name="i" select="position()"/>
                <xsl:choose>
                    <!-- determine the correct spacing and write it -->

```

```

select="concat('(/@j:commentText, 'j:ChargeDispositionDate;', 'j:ChargeDispositionDescriptionText, ')')"/>
select="$amtColSpacer"/>

select="$posMaxWidth"/>

name="WrapedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

name="WrapWidth" select="$posMaxWidth + 1"/>

select="0"/>

select="concat(., '(/@j:reportedDate;', 'j:ChargeDispositionDescriptionText, ')')"/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>

<xsl:when test="$i = 1">
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"
    <xsl:with-param name="StartPos"

    <!-- space over spacer width -->
    <xsl:with-param name="EndPos"

    <xsl:with-param

    <xsl:with-param name="CurrentPos"

    <xsl:with-param

    <xsl:with-param name="initialRun"

  </xsl:call-template>
  <xsl:text>#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over to Column2 start pos

    <xsl:with-param name="EndPos"

```

```

name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test="j:ActivityCommentText">
<xsl:text> Comment</xsl:text>
</xsl:if>
<xsl:for-each select="j:ActivityCommentText">
<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->
<xsl:when test="$i = 1">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over spacer width -->
<xsl:with-param name="EndPos"
<xsl:with-param
select="normalize-space()"/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>

```

```

select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

select="."/>

```

```

select="$posCol2"/>

```

```

-->

```

```

select="$posMaxWidth"/>

```

```

name="WrapedStartPos" select="$posCol2"/>

```

```

select="$posMaxWidth + 1"/>

```

```

name="WrapWidth" select="$posMaxWidth + 1"/>

```

```

select="0"/>

```

```

<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>

```

```

        </xsl:for-each>
    </xsl:for-each>
    <!-- End of Charge Loop -->
</xsl:when>
</xsl:choose>
<!-- Sentencing Section -->
<xsl:choose>
    <xsl:when test="count(rap:Sentencing) > 0">
        <xsl:text>-----
&#10;</xsl:text>
        <xsl:text>Sentencing      (Cycle </xsl:text>
<xsl:value-of select="format-number($Cycle-number,'#000')"/>
<xsl:text>)</xsl:text>
<xsl:text>&#10;</xsl:text>
        <!-- Edited 05/03 -->
<xsl:if test="/rap:Sentencing/j:ActivityDate">
    <xsl:text>Sentence Date</xsl:text>
    <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"
select="/rap:Sentencing/j:ActivityDate"/>
        <xsl:with-param name="StartPos" select="$posCol2 -
(13)"/>
        <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
        <xsl:with-param name="WrappedStartPos"
select="$posCol2"/>
        <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
        <xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>
        <xsl:with-param name="initialRun" select="0"/>
    
```

```

        </xsl:call-template>
        <xsl:text>&#10;</xsl:text>
    </xsl:if>
    <!-- -->
    <!-- Edited 05/03 -->
    <xsl:if test="rap:Sentencing/j:Court">
        <xsl:text>Sentencing Agency</xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text"
select="concat(rap:Sentencing/j:Court/j:OrganizationName, ', rap:Sentencing/j:OrganizationORIID/j:ID)"/>
            <xsl:with-param name="StartPos" select="$posCol2 -
17"/>

            <!-- space over to Column2 start pos -->
            <xsl:with-param name="EndPos"

            <xsl:with-param name="WrappedStartPos"

            <xsl:with-param name="CurrentPos"

            <xsl:with-param name="WrapWidth"

            <xsl:with-param name="initialRun" select="0"/>
        </xsl:call-template>
        <xsl:text>&#10;</xsl:text>
    </xsl:if>
    <xsl:if test="rap:Sentencing/rap:CourtRecordID">
        <xsl:text>Court Case Number</xsl:text>
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text"
select="rap:Sentencing/rap:CourtRecordID/j:ID"/>

```

```

17"/>

select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

<xsl:with-param name="StartPos" select="$posCol2 -
<!-- space over to Column2 start pos -->
<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<!-- -->
<xsl:for-each select="/rap:Sentencing/j:Sentence">
  <!-- Loop through the Sentencings -->
  <!-- -->
  <xsl:for-each select="j:SentenceCharge">
    <!-- Removed Dashes 09/07 -->
    <!-- Edited 05/03 -->
    <xsl:if test="j:ChargeSequenceID/j:ID">
      <xsl:text>Charge </xsl:text>
      <xsl:value-of select="j:ChargeSequenceID/j:ID"/>
      <xsl:text>#10;</xsl:text>
    </xsl:if>
    <xsl:if test="j:ChargeID/j:ID">
      <xsl:text> Charge Number </xsl:text>
      <xsl:value-of select="j:ChargeID/j:ID"/>
      <xsl:text>#10;</xsl:text>

```

```
select="j:ChargeDescriptionText"/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
</xsl:if>  
<xsl:if test="j:ChargeTrackingID">  
  <xsl:text>Charge Tracking Number </xsl:text>  
  <xsl:value-of select="j:ChargeTrackingID/j:ID"/>  
  <xsl:text>#10;</xsl:text>  
</xsl:if>  
<xsl:if test="j:ChargeText">  
  <xsl:text>  Charge Literal </xsl:text>  
  <xsl:value-of select="j:ChargeText"/>  
  <xsl:text>#10;</xsl:text>  
</xsl:if>  
<xsl:if test="j:ChargeDescriptionText">  
  <xsl:text>  Charge Description</xsl:text>  
  <xsl:call-template name="wrapIn">  
    <xsl:with-param name="Text"  
    <xsl:with-param name="StartPos"  
    <xsl:with-param name="EndPos"  
    <xsl:with-param name="WrapedStartPos"  
    <xsl:with-param name="CurrentPos"  
    <xsl:with-param name="WrapWidth"  
    <xsl:with-param name="initialRun" select="0"/>  
  </xsl:call-template>  
  <xsl:text>#10;</xsl:text>  
</xsl:if>  
<xsl:if test="./j:ChargeStatute">
```



```

        <xsl:text> Statute</xsl:text>
    </xsl:if>
    <xsl:for-each select=".j:ChargeStatute">
        <xsl:variable name="i" select="position()"/>
        <xsl:choose>
            <!-- determine the correct spacing and write it -->
            <xsl:when test="$i = 1">
                <xsl:call-template name="wrapIn">
                    <xsl:with-param name="Text"
select="concat(/j:StatuteText, ' (, /j:StatuteCodeID/j:ID, ', ', /j:StatuteJurisdiction/j:LocationStateName, ')')"/>
                    <xsl:with-param name="StartPos"
select="$amtColSpacer"/>
                    <!-- space over spacer width -->
                    <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
                    <xsl:with-param
name="WrapedStartPos" select="$posCol2"/>
                    <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
                    <xsl:with-param
name="WrapWidth" select="$posMaxWidth + 1"/>
                    <xsl:with-param name="initialRun"
select="0"/>
                </xsl:call-template>
                <xsl:text>#10;</xsl:text>
            </xsl:when>
            <xsl:otherwise>
                <xsl:call-template name="wrapIn">
                    <xsl:with-param name="Text"
select="."/>
                </xsl:call-template>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:for-each>

```

```

select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"

</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test=". /j:ChargeReporting/j:ChargeNCICCode">
<xsl:text> NCIC Offense Code </xsl:text>
</xsl:if>
<xsl:for-each select=". /j:ChargeReporting/j:ChargeNCICCode">
<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->
<xsl:when test="$i = 1">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"

select="."/>

```

```

select="$amtColSpacer - 3"/>

select="$posMaxWidth"/>

name="WrapedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

name="WrapWidth" select="$posMaxWidth + 1"/>

select="0"/>

select="."/>

select="$posCol2"/>

-->

select="$posMaxWidth"/>

name="WrapedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

```

```

<xsl:with-param name="StartPos"

<!-- space over spacer width -->
<xsl:with-param name="EndPos"

<xsl:with-param

<xsl:with-param name="CurrentPos"

<xsl:with-param

<xsl:with-param name="initialRun"

</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

  <xsl:with-param name="StartPos"

  <!-- space over to Column2 start pos

  <xsl:with-param name="EndPos"

  <xsl:with-param

  <xsl:with-param name="CurrentPos"

```



```

select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

```

```

select="."/>

```

```

select="$posCol2"/>

```

```

-->

```

```

select="$posMaxWidth"/>

```

```

name="WrapedStartPos" select="$posCol2"/>

```

```

select="$posMaxWidth + 1"/>

```

```

name="WrapWidth" select="$posMaxWidth + 1"/>

```

```

select="0"/>

```

```

<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>

```

```
select=".j:ChargeCountQuantity"/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select=".j:ChargeClassification/j:ChargeSeverityText"/>
```

```
select="$amtColSpacer"/>
```

```
</xsl:for-each>  
<xsl:if test=".j:ChargeStatute/j:StatueOffenseID &lt; 1">  
  <xsl:text>#10;</xsl:text>  
</xsl:if>  
<xsl:if test=".j:ChargeCountQuantity">  
  <xsl:text>Counts</xsl:text>  
  <xsl:call-template name="wrapIn">  
    <xsl:with-param name="Text"  
  
    <xsl:with-param name="StartPos"  
  
    <xsl:with-param name="EndPos"  
  
    <xsl:with-param name="WrapedStartPos"  
  
    <xsl:with-param name="CurrentPos"  
  
    <xsl:with-param name="WrapWidth"  
  
    <xsl:with-param name="initialRun" select="0"/>  
  </xsl:call-template>  
  <xsl:text>#10;</xsl:text>  
</xsl:if>  
<xsl:if test=".j:ChargeClassification/j:ChargeSeverityText">  
  <xsl:text>Severity</xsl:text>  
  <xsl:call-template name="wrapIn">  
    <xsl:with-param name="Text"  
  
    <xsl:with-param name="StartPos"  
  
  <!-- space over spacer width -->
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="./j:ChargeClassification/j:ChargeApplicabilityText"/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun" select="0"/>
```

```
</xsl:call-template>
```

```
<xsl:text>#10;</xsl:text>
```

```
</xsl:if>
```

```
<xsl:if test="./j:ChargeClassification/j:ChargeApplicabilityText">
```

```
<xsl:text> Inchoate Charge</xsl:text>
```

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos"
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun" select="0"/>
```

```
</xsl:call-template>
```

```
<xsl:text>#10;</xsl:text>
```

```
</xsl:if>
```

```

test=".j:ChargeClassification/j:ChargeSpecialAllegationText">
    <xsl:if
        <xsl:text> Enhancing Factor</xsl:text>
    </xsl:if>
    <xsl:for-each
        <xsl:variable name="i" select="position()"/>
        <xsl:choose>
            <!-- determine the correct spacing and write it -->
            <xsl:when test="$i = 1">
                <xsl:call-template name="wrapIn">
                    <xsl:with-param name="Text"
                        <xsl:with-param name="StartPos"
                            <!-- space over spacer width -->
                            <xsl:with-param name="EndPos"
                                <xsl:with-param
                                    <xsl:with-param name="CurrentPos"
                                        <xsl:with-param
                                            <xsl:with-param name="initialRun"
                                                </xsl:call-template>
                                                <xsl:text>&#10;</xsl:text>
                                            </xsl:when>
                                            <xsl:otherwise>
                                                <xsl:call-template name="wrapIn">

```



```

select="."/>

select="$posCol2"/>

-->

select="$posMaxWidth"/>

name="WrapedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

name="WrapWidth" select="$posMaxWidth + 1"/>

select="0"/>

test="./j:ChargeClassification/j:ChargeReducingFactorText">

select="./j:ChargeClassification/j:ChargeReducingFactorText">

```

```

<xsl:with-param name="Text"

<xsl:with-param name="StartPos"

<!-- space over to Column2 start pos

<xsl:with-param name="EndPos"

<xsl:with-param

<xsl:with-param name="CurrentPos"

<xsl:with-param

<xsl:with-param name="initialRun"

</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if

<xsl:text> Reducing Factor</xsl:text>
</xsl:if>
<xsl:for-each

<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->
<xsl:when test="$i = 1">

```

```

select="."/>

select="$amtColSpacer"/>

select="$posMaxWidth"/>

name="WraPedStartPos" select="$posCol2"/>

select="$posMaxWidth + 1"/>

name="WrapWidth" select="$posMaxWidth + 1"/>

select="0"/>

```

```

select="."/>

select="$posCol2"/>

-->

select="$posMaxWidth"/>

name="WraPedStartPos" select="$posCol2"/>

```

```

<xsl:call-template name="wrapIn">
  <xsl:with-param name="Text"

  <xsl:with-param name="StartPos"

  <!-- space over spacer width -->
  <xsl:with-param name="EndPos"

  <xsl:with-param

  <xsl:with-param name="CurrentPos"

  <xsl:with-param

  <xsl:with-param name="initialRun"

</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"

    <xsl:with-param name="StartPos"

    <!-- space over to Column2 start pos

    <xsl:with-param name="EndPos"

    <xsl:with-param

```

```

select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test=".j:ChargeDisposition">
<xsl:text> Disposition</xsl:text>
</xsl:if>
<xsl:for-each select=".j:ChargeDisposition">
<xsl:variable name="i" select="position()"/>
<xsl:choose>
<!-- determine the correct spacing and write it -->
<xsl:when test="$i = 1">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over spacer width -->
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
select="concat('(./@j:commentText, 'j:ChargeDispositionDate;', 'j:ChargeDispositionDescriptionText, ')')"/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>

```

```

name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

select="concat(., ' (./@j:reportedDate, '; 'j:ChargeDispositionDescriptionText, ')')"/>
select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>

<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<xsl:if test="j:ActivityCommentText">

```

```

select="normalize-space(.)"/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>
select="."/>

```

```

<xsl:text>          Comment</xsl:text>
</xsl:if>
<xsl:for-each select="j:ActivityCommentText">
  <xsl:variable name="i" select="position()"/>
  <xsl:choose>
    <!-- determine the correct spacing and write it -->
    <xsl:when test="$i = 1">
      <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text"
          <xsl:with-param name="StartPos"
            <!-- space over spacer width -->
            <xsl:with-param name="EndPos"
              <xsl:with-param
                <xsl:with-param name="CurrentPos"
                  <xsl:with-param
                    <xsl:with-param name="initialRun"
                      </xsl:call-template>
                      <xsl:text>&#10;</xsl:text>
                    </xsl:when>
                    <xsl:otherwise>
                      <xsl:call-template name="wrapIn">
                        <xsl:with-param name="Text"

```

```

select="$posCol2"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
</xsl:for-each>
<!-- End of Charge Loop -->
<xsl:choose>
<xsl:when test="count(j:SentenceDescriptionText) &lt; 1">
<xsl:text>#10;</xsl:text>
</xsl:when>
<xsl:otherwise>
<xsl:for-each select="j:SentenceDescriptionText">
<xsl:text>Sentence</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text"
select="."/>

```

```

select="$posCol2 - 8"/>
-->
select="$posMaxWidth"/>
name="WrapedStartPos" select="$posCol2"/>
select="$posMaxWidth + 1"/>
name="WrapWidth" select="$posMaxWidth + 1"/>
select="0"/>
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos
<xsl:with-param name="EndPos"
<xsl:with-param
<xsl:with-param name="CurrentPos"
<xsl:with-param
<xsl:with-param name="initialRun"
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
</xsl:otherwise>
</xsl:choose>
</xsl:for-each>
<!-- End of Sentencing Loop -->
</xsl:when>
</xsl:choose>
<!-- Corrections Section -->
<xsl:choose>
<xsl:when test="count(rap:Supervision) > 0">
<xsl:text>-----
#10;</xsl:text>
<xsl:text>Corrections (Cycle </xsl:text>
<xsl:value-of select="format-number($Cycle-number, '#000')"/>
<xsl:text>)</xsl:text>

```

```

<xsl:text>&#10;</xsl:text>
<!-- Edited 05/03 -->
<xsl:if test="rap:Supervision/j:SupervisionAgency">
  <xsl:text>Corrections Agency</xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"
select="concat(rap:Supervision/j:SupervisionAgency/j:OrganizationName,'
'j:Supervision/j:SupervisionAgency/j:OrganizationORIID/j:ID)"/>
    <xsl:with-param name="StartPos" select="$posCol2 -
18"/>
    <!-- space over to Column2 start pos -->
    <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
    <xsl:with-param name="WrappedStartPos"
select="$posCol2"/>
    <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
    <xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>
    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
  <xsl:text>&#10;</xsl:text>
</xsl:if>
<!-- -->
<!-- Edited 05/03 -->
<xsl:if test="rap:Supervision/j:SupervisionSubject/j:PersonName">
  <xsl:text>Subject's Name</xsl:text>
</xsl:if>
<!-- -->
<xsl:for-each
select="rap:Supervision/j:SupervisionSubject/j:PersonName">

```



```

data -->
select="$posCol2 - 14"/>
select="0"/>
select="$posCol2"/>
select="0"/>
<!-- Name -->
<xsl:variable name="i" select="position()"/>
<xsl:choose>
  <!-- determine the correct spacing before the write of name
  <xsl:when test="$i = 1">
    <xsl:call-template name="spaceover">
      <xsl:with-param name="amount"
      <xsl:with-param name="amountWrote"
    </xsl:call-template>
  </xsl:when>
  <xsl:otherwise>
    <xsl:call-template name="spaceover">
      <xsl:with-param name="amount"
      <xsl:with-param name="amountWrote"
    </xsl:call-template>
  </xsl:otherwise>
</xsl:choose>
<xsl:value-of select="./j:PersonFullName"/>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
<xsl:if test="not(j:Supervision/j:SupervisionSubject/j:PersonName)">
  <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:for-each select="rap:Supervision">
  <!-- Loop through the Corrections -->
  <!-- Edited 05/03 -->

```

```
select="j:SupervisionSubject/j:SubjectID/j:ID"/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="j:Supervision/j:CaseTrackingID/j:ID"/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
<xsl:if test="j:SupervisionSubject/j:SubjectID">  
  <xsl:text>Correctional Id Number</xsl:text>  
  <xsl:call-template name="wrapIn">  
    <xsl:with-param name="Text"  
  
    <xsl:with-param name="StartPos"  
  
    <xsl:with-param name="EndPos"  
  
    <xsl:with-param name="WrapedStartPos"  
  
    <xsl:with-param name="CurrentPos"  
  
    <xsl:with-param name="WrapWidth"  
  
    <xsl:with-param name="initialRun" select="0"/>  
  </xsl:call-template>  
  <xsl:text>&#10;</xsl:text>  
</xsl:if>  
<!-- -->  
<!-- Edited 05/03 -->  
<xsl:if test="j:Supervision/j:CaseTrackingID/j:ID">  
  <xsl:text> Court Case Number</xsl:text>  
  <xsl:call-template name="wrapIn">  
    <xsl:with-param name="Text"  
  
    <xsl:with-param name="StartPos"  
  
    <xsl:with-param name="EndPos"
```

```

select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<!-- -->
<xsl:for-each select="j:SupervisionCustodyStatus">
  <xsl:variable name="i" select="position()"/>
  <xsl:text> Correction Action</xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text"
    <xsl:with-param name="StartPos"
    <!-- space over to Column2 start pos -->
    <xsl:with-param name="EndPos"
    <xsl:with-param name="WrapedStartPos"
    <xsl:with-param name="CurrentPos"
    <xsl:with-param name="WrapWidth"
    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
  <xsl:text>#10;</xsl:text>

```

```
space."/>
```

```
select="$amtColSpacer"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
space."/>
```

```
select="$amtColSpacer"/>
```

```
</xsl:for-each>  
<xsl:for-each select="j:ActivityCommentText">  
  <xsl:variable name="i" select="position()"/>  
  <xsl:text> Correction Comment</xsl:text>  
  <xsl:call-template name="wrapIn">  
    <xsl:with-param name="Text" select="normalize-  
  
    <xsl:with-param name="StartPos"  
  
    <!-- space over to Column2 start pos -->  
    <xsl:with-param name="EndPos"  
  
    <xsl:with-param name="WrapedStartPos"  
  
    <xsl:with-param name="CurrentPos"  
  
    <xsl:with-param name="WrapWidth"  
  
    <xsl:with-param name="initialRun" select="0"/>  
  </xsl:call-template>  
  <xsl:text>#10;</xsl:text>  
</xsl:for-each>  
<xsl:for-each select="j:ActivityTypeText">  
  <xsl:variable name="i" select="position()"/>  
  <xsl:text>Sentence Type </xsl:text>  
  <xsl:call-template name="wrapIn">  
    <xsl:with-param name="Text" select="normalize-  
  
    <xsl:with-param name="StartPos"  
  
    <!-- space over to Column2 start pos -->
```

```

select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

space."/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

```

```

<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
<xsl:for-each select="j:ActivityDescriptionText">
<xsl:variable name="i" select="position()"/>
<xsl:text>Sentence Description </xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text" select="normalize-
<xsl:with-param name="StartPos"
<!-- space over to Column2 start pos -->
<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>

```

```

space."/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
<!--
select="$amtColSpacer"/>
<xsl:text>&#10;</xsl:text>
</xsl:for-each>
<xsl:for-each select="j:ActivityReasonText">
  <xsl:variable name="i" select="position()"/>
  <xsl:text>Charge Description </xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text" select="normalize-
    <xsl:with-param name="StartPos"
    <!-- space over to Column2 start pos -->
    <xsl:with-param name="EndPos"
    <xsl:with-param name="WrappedStartPos"
    <xsl:with-param name="CurrentPos"
    <xsl:with-param name="WrapWidth"
    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
  <xsl:text>&#10;</xsl:text>
</xsl:for-each>
<xsl:for-each select="j:SupervisionRelease">
  <xsl:variable name="i" select="position()"/>
  <xsl:text>Release Type </xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text" select="."/>
    <xsl:with-param name="StartPos"
    over to Column2 start pos

```

```

select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

```

```
-->
```

```

space(/j:ActivityDate)/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>

```

```

<xsl:with-param name="EndPos"
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
<xsl:text>Release Date </xsl:text>
<xsl:call-template name="wrapIn">
  <xsl:with-param name="Text" select="normalize-
  <xsl:with-param name="StartPos"
  <!-- space over to Column2 start pos -->
  <xsl:with-param name="EndPos"
  <xsl:with-param name="WrapedStartPos"
  <xsl:with-param name="CurrentPos"
  <xsl:with-param name="WrapWidth"
  <xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:for-each>

```

```

space(/j:ActivityTypeText)/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
space(/j:ActivityDate)/>
select="$amtColSpacer"/>
select="$posMaxWidth"/>

```

```

<xsl:for-each select="j:SupervisionConditionalRelease">
  <xsl:variable name="i" select="position()"/>
  <xsl:text>Cond. Release Type </xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text" select="normalize-

    <xsl:with-param name="StartPos"

    <!-- space over to Column2 start pos -->
    <xsl:with-param name="EndPos"

    <xsl:with-param name="WrappedStartPos"

    <xsl:with-param name="CurrentPos"

    <xsl:with-param name="WrapWidth"

    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
  <xsl:text>&#10;</xsl:text>

  <xsl:text>Cond. Release Date </xsl:text>
  <xsl:call-template name="wrapIn">
    <xsl:with-param name="Text" select="normalize-

    <xsl:with-param name="StartPos"

    <!-- space over to Column2 start pos -->
    <xsl:with-param name="EndPos"

```



```

select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
</xsl:for-each>
<!-- End of Correction Loop -->
</xsl:when>
</xsl:choose>
</xsl:for-each>
<!-- End of the Cycle Loop -->
<!-- ##### Index of Agencies ##### -->
<xsl:text>***** INDEX OF AGENCIES
*****&#10;&#10;</xsl:text>
<xsl:for-each select="//rap:RapSheet/j:Organization">
<xsl:text>Agency</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text" select="concat(j:OrganizationName,
'j:OrganizationORIID/j:ID,;')"/>
<xsl:with-param name="StartPos" select="$posCol2 - 6"/>
<!-- space over to Column2 start pos -->
<xsl:with-param name="EndPos" select="$posMaxWidth"/>
<xsl:with-param name="WrapedStartPos" select="$posCol2"/>
<xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
<xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>

```

```

        <xsl:with-param name="initialRun" select="0"/>
    </xsl:call-template>
    <xsl:text>#10;</xsl:text>
    <xsl:if test="count(j:OrganizationUnitName) > 0">
        <xsl:text>Contact        </xsl:text>
        <xsl:value-of select="j:OrganizationUnitName"/>
        <xsl:text> Department</xsl:text>
        <xsl:text>#10;</xsl:text>
    </xsl:if>
    <xsl:for-each
select="j:OrganizationLocation/j:LocationContactInformation/j:ContactInformationDescriptionText">
        <xsl:call-template name="wrapIn">
            <xsl:with-param name="Text" select="."/>
            <xsl:with-param name="StartPos" select="$posCol2"/>
            <!-- space over to Column2 start pos -->
            <xsl:with-param name="EndPos" select="$posMaxWidth"/>
            <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
            <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
            <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
            <xsl:with-param name="initialRun" select="0"/>
        </xsl:call-template>
        <xsl:text>#10;</xsl:text>
    </xsl:for-each>
    <!-- Telephone -->
    <xsl:choose>
        <xsl:when
test="j:OrganizationLocation/j:LocationContactInformation/j:ContactTelephoneNumber/j:TelephoneNumberFullIID > 0">
            <!-- if there is a telephone number, write it -->
            <xsl:for-each
select="j:OrganizationLocation/j:LocationContactInformation/j:ContactTelephoneNumber">
                <xsl:text>Agency Telephone</xsl:text>
            </xsl:for-each>
        </xsl:when>
    </xsl:choose>

```

```
0">
```

```
-->
```

```
select="j:TelephoneNumberFullID"/>
```

```
select="$posCol2 - 16"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="0"/>
```

```
select="concat(j:TelephoneNumberFullID"/>
```

```
select="$posCol2 - 16"/>
```

```
select="$posMaxWidth"/>
```

```
<xsl:choose>
```

```
<xsl:when test="count(j:TelephoneNumberFullID) &gt;
```

```
<!-- if standard phone format, format it accordingly
```

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos"
```

```
<xsl:with-param name="EndPos"
```

```
<xsl:with-param name="WrapedStartPos"
```

```
<xsl:with-param name="CurrentPos"
```

```
<xsl:with-param name="WrapWidth"
```

```
<xsl:with-param name="initialRun"
```

```
<!-- 0=True -->
```

```
</xsl:call-template>
```

```
</xsl:when>
```

```
<xsl:otherwise>
```

```
<xsl:call-template name="wrapIn">
```

```
<xsl:with-param name="Text"
```

```
<xsl:with-param name="StartPos"
```

```
<xsl:with-param name="EndPos"
```

```

select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
select="0"/>
<xsl:with-param name="WrapedStartPos"
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun"
<!-- 0=True -->
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
</xsl:when>
<xsl:otherwise>
<!-- xsl:text>#10;</xsl:text -->
</xsl:otherwise>
</xsl:choose>
<!-- FAX -->
<xsl:choose>
<xsl:when
test="count(//rap:RapSheet/j:Organization/j:OrganizationLocation/j:LocationContactInformation/j:ContactFacsimileNumber) >
0">
<!-- if there is a telephone number, write it -->
<xsl:for-each
select="//rap:RapSheet/j:Organization/j:OrganizationLocation/j:LocationContactInformation/j:ContactFacsimileNumber">
<xsl:text>Agency Facsimile</xsl:text>
<xsl:choose>
<xsl:when test="count(j:TelephoneNumberFullID) >
0">

```

```
-->
```

```
select="j:TelephoneNumberFullID"/>
```

```
select="$posCol2 - 16"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="$posMaxWidth + 1"/>
```

```
select="0"/>
```

```
select="j:TelephoneNumberFullID"/>
```

```
select="$posCol2 - 16"/>
```

```
select="$posMaxWidth"/>
```

```
select="$posCol2"/>
```

```
<!-- if standard phone format, format it accordingly
```

```
<xsl:call-template name="wrapIn">  
  <xsl:with-param name="Text"
```

```
  <xsl:with-param name="StartPos"
```

```
  <xsl:with-param name="EndPos"
```

```
  <xsl:with-param name="WrapedStartPos"
```

```
  <xsl:with-param name="CurrentPos"
```

```
  <xsl:with-param name="WrapWidth"
```

```
  <xsl:with-param name="initialRun"
```

```
  <!-- 0=True -->
```

```
  </xsl:call-template>
```

```
</xsl:when>
```

```
<xsl:otherwise>
```

```
  <xsl:call-template name="wrapIn">
```

```
    <xsl:with-param name="Text"
```

```
    <xsl:with-param name="StartPos"
```

```
    <xsl:with-param name="EndPos"
```

```
    <xsl:with-param name="WrapedStartPos"
```

```

select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
select="0"/>
<xsl:with-param name="CurrentPos"
<xsl:with-param name="WrapWidth"
<xsl:with-param name="initialRun"
<!-- 0=True -->
</xsl:call-template>
</xsl:otherwise>
</xsl:choose>
<xsl:text>#10;</xsl:text>
</xsl:for-each>
</xsl:when>
<xsl:otherwise>
<!-- xsl:text>#10;</xsl:text -->
</xsl:otherwise>
</xsl:choose>
<xsl:for-each
select="//rap:RapSheet/j:Organization/j:OrganizationLocation/j:LocationContactInformation/j:ContactEmailID/j:ID">
<xsl:variable name="i" select="position()"/>
<xsl:text>Agency Email Address</xsl:text>
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text" select="."/>
<xsl:with-param name="StartPos" select="$posCol2 - 20"/>
<!-- space over to Column2 start pos -->
<xsl:with-param name="EndPos" select="$posMaxWidth"/>
<xsl:with-param name="WrappedStartPos" select="$posCol2"/>
<xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
<xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>

```

```

        <xsl:text>&#10;</xsl:text>
    </xsl:for-each>
    <!-- xsl:if test="count(email) &lt; 1"><xsl:text>&#10;</xsl:text></xsl:if -->
    <!-- Address -->
    <xsl:for-each select=".j:OrganizationLocation">
        <xsl:if test="j:LocationAddress">
            <xsl:text>Address</xsl:text>
        </xsl:if>
        <xsl:if test=".j:LocationAddress/j:BuildingName">
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select=".j:LocationAddress/j:BuildingName"/>
                <xsl:with-param name="StartPos" select="$posCol2 - 8"/>
                <xsl:with-param name="EndPos" select="$posMaxWidth"/>
                <xsl:with-param name="WrappedStartPos" select="$posCol2"/>
                <xsl:with-param name="CurrentPos" select="$posMaxWidth +
1"/>
                <xsl:with-param name="WrapWidth" select="$posMaxWidth +
1"/>
                <xsl:with-param name="initialRun" select="0"/>
            </xsl:call-template>
        </xsl:if>
        <xsl:text>&#10;</xsl:text>
        <xsl:if test=".j:LocationAddress/j:LocationStreet/j:StreetFullText">
            <xsl:call-template name="wrapIn">
                <xsl:with-param name="Text"
select=".j:LocationAddress/j:LocationStreet/j:StreetFullText"/>
                <xsl:with-param name="StartPos" select="$posCol2"/>
                <xsl:with-param name="EndPos" select="$posMaxWidth"/>
                <xsl:with-param name="WrappedStartPos" select="$posCol2"/>

```

```

1"/>
1"/>
<xsl:with-param name="CurrentPos" select="$posMaxWidth +
<xsl:with-param name="WrapWidth" select="$posMaxWidth +
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="count(/j:LocationAddress/j:LocationStreet/j:StreetFullText) = 0">
  <xsl:if test="./j:LocationAddress/j:LocationStreet/j:StreetName">
    <xsl:call-template name="wrapIn">
      <xsl:with-param name="Text"
select="concat(/j:LocationAddress/j:LocationStreet/j:StreetNumberText,'
./j:LocationAddress/j:LocationStreet/j:StreetPredirectionalText','./j:LocationAddress/j:LocationStreet/j:StreetName,'
./j:LocationAddress/j:LocationStreet/j:StreetTypeText)"/>
    <xsl:with-param name="StartPos" select="$posCol2"/>
    <xsl:with-param name="EndPos"
select="$posMaxWidth"/>
    <xsl:with-param name="WrappedStartPos"
select="$posCol2"/>
    <xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
    <xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>
    <xsl:with-param name="initialRun" select="0"/>
  </xsl:call-template>
  <xsl:text>#10;</xsl:text>
</xsl:if>
<xsl:if test="./j:LocationAddress/j:LocationStateText">
  <xsl:call-template name="wrapIn">

```



```

select="concat(/j:LocationAddress/j:LocationCityText,' ',./j:LocationAddress/j:LocationStateText,'
',./j:LocationAddress/j:LocationPostalCodeID,' ',./j:LocationAddress/j:LocationCountryText)"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
',./j:LocationAddress/j:LocationCountyName)"/>
select="$posMaxWidth"/>
select="$posCol2"/>
select="$posMaxWidth + 1"/>
select="$posMaxWidth + 1"/>
<xsl:with-param name="Text"
select="concat(/j:LocationAddress/j:LocationCityText,' ',./j:LocationAddress/j:LocationStateText,'
',./j:LocationAddress/j:LocationPostalCodeID,' ',./j:LocationAddress/j:LocationCountryText)"/>
<xsl:with-param name="StartPos" select="$posCol2"/>
<xsl:with-param name="EndPos"
select="$posMaxWidth"/>
<xsl:with-param name="WrapedStartPos"
select="$posCol2"/>
<xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
<xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>
<xsl:text>&#10;</xsl:text>
</xsl:if>
<xsl:if test="./j:LocationAddress/j:LocationCountyName">
<xsl:call-template name="wrapIn">
<xsl:with-param name="Text" select="concat('County:
',./j:LocationAddress/j:LocationCountyName)"/>
<xsl:with-param name="StartPos" select="$posCol2"/>
<xsl:with-param name="EndPos"
select="$posMaxWidth"/>
<xsl:with-param name="WrapedStartPos"
select="$posCol2"/>
<xsl:with-param name="CurrentPos"
select="$posMaxWidth + 1"/>
<xsl:with-param name="WrapWidth"
select="$posMaxWidth + 1"/>
<xsl:with-param name="initialRun" select="0"/>
</xsl:call-template>

```

```

                                <xsl:text>#10;</xsl:text>
                            </xsl:if>
                        </xsl:if>
                    </xsl:for-each>
                <xsl:text>#10;</xsl:text>
                <xsl:if test="position() != last()">
                    <xsl:text>-----
#10;</xsl:text>
                                </xsl:if>
                            </xsl:for-each>
                        <!-- End of Agency Index Loop -->
                    </xsl:otherwise>
                </xsl:choose>
                <xsl:text>* * * END OF RECORD * * * </xsl:text>
            </xsl:template>
            <!-- ##### TEMPLATES and GLOBALS ##### -->
            <xsl:template name="textwrap">
                <!--

```

\*\*\*\*\*

```

|| The below template wraps a text string at a specified position.
|| Input:
||   CurrentPos - Hold Position (wrap position on first call) + 1.
||   WrapWidth - Wrap Width (character position of which to wrap at) + 1.
|| Explanation/Notes:
||   Before calling this template it is recommended you declare a variable
||   (ie WrapAt) and set this variable to the Position of which you want
||   the text string wrapped at. Next set the input parameters BOTH to
||   the variable's value plus one (ie WrapAt + 1).
|| Example Call:
|| <xsl:variable name="WrapAt" select="72"/>
|| <xsl:call-template name="textwrap">

```

```

||      <xsl:with-param name="Text" select="//rap:RapSheet/Introduction/caveat"/>
||      <xsl:with-param name="CurrentPos" select="$WrapAt+1"/>
||      <xsl:with-param name="WrapWidth" select="$WrapAt+1"/>
||  </xsl:call-template>
*****
-->

<xsl:param name="Text"/>
<xsl:param name="CurrentPos"/>
<xsl:param name="WrapWidth"/>
<xsl:choose>
  <xsl:when test="$CurrentPos &lt; 1">
    <!-- If CurrentPos = 0 it means the word is longer than the wrap width
so output as much as possible on one line, then pass the
remainder of the string on for more processing. -->
    <xsl:value-of select="substring($Text, 1, $WrapWidth - 1)"/>
    <xsl:text>&#10;</xsl:text>
    <xsl:call-template name="textwrap">
      <xsl:with-param name="Text" select="substring($Text,$WrapWidth)"/>
      <xsl:with-param name="CurrentPos" select="$WrapWidth"/>
      <xsl:with-param name="WrapWidth" select="$WrapWidth"/>
    </xsl:call-template>
  </xsl:when>
  <xsl:when test="string-length($Text) > ($CurrentPos - 1)">
    <!-- if input string length is greater than 72 wrap it up -->
    <xsl:choose>
      <xsl:when test="substring($Text, $CurrentPos - 1, 1) = ' '">
        <!-- return the string -->
        <xsl:value-of select="substring($Text, 1, $CurrentPos - 1)"/>
        <xsl:text>&#10;</xsl:text>
        <xsl:call-template name="textwrap">
          <xsl:with-param name="Text" select="substring($Text,$CurrentPos)"/>

```

```

        <xsl:with-param name="CurrentPos" select="$WrapWidth"/>
        <xsl:with-param name="WrapWidth" select="$WrapWidth"/>
    </xsl:call-template>
</xsl:when>
<xsl:otherwise>
    <!-- go back a character and reEvaluate -->
    <xsl:call-template name="textwrap">
        <xsl:with-param name="Text" select="$Text"/>
        <xsl:with-param name="CurrentPos" select="$CurrentPos - 1"/>
        <xsl:with-param name="WrapWidth" select="$WrapWidth"/>
    </xsl:call-template>
</xsl:otherwise>
</xsl:choose>
</xsl:when>
<xsl:otherwise>
    <xsl:value-of select="$Text"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
<xsl:template name="wrapIn">
    <!--

```

\*\*\*\*\*

```

|| The below template wraps a text string at a specified position given a specified Start and
|| end position.
|| Input:
|| Text      - The inputted text to wrap: NOTE: perform a normalize-string() before to
||            strip extra spaces and carriage returns.
|| StartPos  - The start position of where the string will start writing from.
|| EndPos    - The end position of where the string will stop writing and begin a wrap.
|| WrappedStartPos - The wrapped start position. Where to start writing on the wrapped line.
|| CurrentPos - Current position (used internally).

```

```

|| WrapWidth    - The wrap width.
|| initialRun   - Initial Run (used internally). Identifies if it is the first run of the
||               template. If it is the first run, the amount to space over before writing
||               can and will be different so this parameter adjusts accordingly.
|| Explanation/Notes:
|| As mentioned in the input section, perform a normalize-string on the inputted text to remove
|| unwanted extra spaces or carriage returns.
|| Example Call:
|| <xsl:call-template name="wrapIn">
||     <xsl:with-param name="Text" select="normalize-space(/example/comments)"/>
||     <xsl:with-param name="StartPos" select="$posCol2"/>
||     <xsl:with-param name="EndPos" select="$posMaxWidth"/>
||     <xsl:with-param name="WrapedStartPos" select="$posCol2"/>
||     <xsl:with-param name="CurrentPos" select="$posMaxWidth + 1"/>
||     <xsl:with-param name="WrapWidth" select="$posMaxWidth + 1"/>
||     <xsl:with-param name="initialRun" select="0"/>
|| </xsl:call-template>
*****
-->
<xsl:param name="Text"/>
<xsl:param name="StartPos"/>
<xsl:param name="EndPos"/>
<xsl:param name="WrapedStartPos"/>
<xsl:param name="CurrentPos"/>
<xsl:param name="WrapWidth"/>
<xsl:param name="initialRun"/>
<xsl:variable name="l" select="$EndPos - $WrapedStartPos"/>
<xsl:variable name="n-text" select="normalize-space($Text)"/>
<xsl:choose>
  <xsl:when test="string-length($Text) <= $l">
    <!-- if passed in string is lt or = the length we can write it -->

```

```

<xsl:choose>
  <!-- Determine the correct spacing before writing -->
  <xsl:when test="$initialRun = 0">
    <!-- 0=True -->
    <xsl:call-template name="spaceover">
      <xsl:with-param name="amount" select="$StartPos"/>
      <!-- use StartPos (adjusted started position) -->
      <xsl:with-param name="amountWrote" select="0"/>
    </xsl:call-template>
  </xsl:when>
  <xsl:otherwise>
    <xsl:call-template name="spaceover">
      <xsl:with-param name="amount" select="$WrappedStartPos"/>
      <!-- if not initial run then it is a wrapped line -->
      <xsl:with-param name="amountWrote" select="0"/>
    </xsl:call-template>
  </xsl:otherwise>
</xsl:choose>
<xsl:value-of select="$Text"/>
<!-- Write the string -->
</xsl:when>
<xsl:otherwise>
  <!-- else wrap the string -->
  <xsl:choose>
    <xsl:when test="$CurrentPos < 1">
      <!-- If CurrentPos = 0 it means the word is longer than the wrap width
      so output as much as possible on one line, then pass the
      remainder of the string on for more processing. -->
      <xsl:text>&#10;</xsl:text>
      <xsl:call-template name="wrapIn">
        <xsl:with-param name="Text" select="substring($Text,$WrapWidth)"/>

```

```

        <xsl:with-param name="StartPos" select="$StartPos"/>
        <xsl:with-param name="EndPos" select="$EndPos"/>
        <xsl:with-param name="WrapedStartPos" select="$WrapedStartPos"/>
        <xsl:with-param name="CurrentPos" select="$WrapWidth"/>
        <xsl:with-param name="WrapWidth" select="$WrapWidth"/>
        <xsl:with-param name="initialRun" select="1"/>
    </xsl:call-template>
</xsl:when>
<xsl:when test="(string-length($Text) + $WrapedStartPos) > ($CurrentPos - 1)">
    <!-- if input string length is greater than 72 wrap it up -->
    <xsl:choose>
        <xsl:when test="substring($Text, (($CurrentPos - 1)- $WrapedStartPos), 1) = ' '">
            <!-- if the last character in the string is a [space] then write the line else
                move back and retry -->

            <xsl:choose>
                <!-- get the correct spacing -->
                <xsl:when test="$initialRun = 0">
                    <!-- 0=True -->
                    <xsl:call-template name="spaceover">
                        <xsl:with-param name="amount"

                                <!-- use StartPos (adjusted started position) -->
                                <xsl:with-param name="amountWrote"

                                        </xsl:call-template>
                                        </xsl:when>
                                        <xsl:otherwise>
                                            <xsl:call-template name="spaceover">
                                                <xsl:with-param name="amount"

                                                        <!-- if not initial run then it is a wrapped line -->

```

```

select="0"/>
                                <xsl:with-param name="amountWrote"
                                </xsl:call-template>
                                </xsl:otherwise>
                                </xsl:choose>
                                <xsl:value-of select="substring($Text, 1, ($CurrentPos - 1) -
$WrpapedStartPos)"/>
                                <xsl:text>&#10;</xsl:text>
                                <xsl:call-template name="wrapIn">
                                <xsl:with-param name="Text"
                                <xsl:with-param name="StartPos" select="$StartPos"/>
                                <xsl:with-param name="EndPos" select="$EndPos"/>
                                <xsl:with-param name="WrpapedStartPos"
                                <xsl:with-param name="CurrentPos" select="$WrapWidth"/>
                                <xsl:with-param name="WrapWidth" select="$WrapWidth"/>
                                <xsl:with-param name="initialRun" select="1"/>
                                <!-- set to 1 as we wrote the above line -->
                                </xsl:call-template>
                                </xsl:when>
                                <xsl:otherwise>
                                <!-- go back a character and reEvaluate -->
                                <xsl:call-template name="wrapIn">
                                <xsl:with-param name="Text" select="$Text"/>
                                <xsl:with-param name="StartPos" select="$StartPos"/>
                                <xsl:with-param name="EndPos" select="$EndPos"/>
                                <xsl:with-param name="WrpapedStartPos"
                                <xsl:with-param name="CurrentPos" select="$CurrentPos - 1"/>
                                <xsl:with-param name="WrapWidth" select="$WrapWidth"/>
select="$WrpapedStartPos"/>

```



```

        <xsl:with-param name="initialRun" select="$initialRun"/>
        <!-- leave at inputted value as we have not wrote a line -->
    </xsl:call-template>
    </xsl:otherwise>
  </xsl:choose>
</xsl:when>
<xsl:otherwise>
  <xsl:value-of select="$Text"/>
</xsl:otherwise>
</xsl:choose>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
<xsl:template name="spaceover">
  <!--
*****
|| Adds a given number of spaces to the output. This template will write out (return) a given
|| number of spaces. Chage the amount to the desired number of spaces you wish to write out.
|| Input:
|| amount - Number of spaces to write out.
|| amountWrote - Hold Variable: Used to signafy amount wrote, start with 0.
|| Example Call:
|| <xsl:call-template name="spaceover">
||     <xsl:with-param name="amount" select="10"/>
||     <xsl:with-param name="amountWrote" select="0"/>
|| </xsl:call-template>
*****
-->
  <xsl:param name="amount"/>
  <xsl:param name="amountWrote"/>
  <xsl:if test="$amountWrote &lt; $amount">

```

```

        <xsl:text> </xsl:text>
        <xsl:call-template name="spaceover">
            <xsl:with-param name="amount" select="$amount"/>
            <xsl:with-param name="amountWrote" select="$amountWrote + 1"/>
        </xsl:call-template>
    </xsl:if>
</xsl:template>
<xsl:template name="assignvalue">
<!--
*****
|| Assigns Value to variable.
|| Input:
|| flag      - variable.
|| flagvalue - value.
|| Example Call:
|| <xsl:call-template name="assignvalue">
||     <xsl:with-param name="flag" select="0"/>
||     <xsl:with-param name="flagvalue" select="1"/>
|| </xsl:call-template>
*****
-->
    <xsl:param name="flag"/>
    <xsl:param name="flagvalue"/>
    <xsl:if test="$flag &lt; $flagvalue">
        <xsl:call-template name="assignvalue">
            <xsl:with-param name="flag" select="$flag + 1"/>
            <xsl:with-param name="flagvalue" select="$flagvalue"/>
        </xsl:call-template>
    </xsl:if>
</xsl:template>
<xsl:template name="doAgency">

```

```

<!--
*****
|| Writes the appropriate 'agency' info (ori or entityName). If future changes require structure
|| changes to the agency write-out format it can easily be accomodated here.
|| Input:
|| agency: entityName, ori, entityAbbreviatedName, and entityAccronym.
|| Example Call:
|| <xsl:call-template name="doAgency">
||     <xsl:with-param name="entityName" select="/agency/entityName"/>
||     <xsl:with-param name="ori" select="/agency/ori"/>
||     :with-param name="entityAbbreviatedName" select="/agency/entityAbbreviatedName"/>
||     <xsl:with-param name="entityAccronym" select="/agency/entityAccronym"/>
|| </xsl:call-template>
*****
-->
<xsl:param name="entityName"/>
<xsl:param name="ori"/>
<xsl:param name="entityAbbreviatedName"/>
<xsl:param name="entityAccronym"/>
<xsl:choose>
  <xsl:when test="count($ori)>0">
    <xsl:value-of select="$ori"/>
  </xsl:when>
  <xsl:when test="count($entityName)>0">
    <xsl:value-of select="$entityName"/>
  </xsl:when>
  <xsl:otherwise/>
</xsl:choose>
</xsl:template>
<xsl:template name="doSMTCode">
  <!--

```

```

*****
|| Writes appropriate SMT Code Section.
|| Input:
||   codeSource
|| Example Call:
||   <xsl:call-template name="doSMTCode">
||     <xsl:with-param name="codeSource" select="."/>
||   </xsl:call-template>
*****
-->
  <xsl:param name="codeSource"/>
  <xsl:choose>
    <xsl:when test="string-length($codeSource)>0">
      <xsl:value-of select="$codeSource"/>
    </xsl:when>
    <xsl:otherwise>
      <xsl:text>Unknown Code</xsl:text>
    </xsl:otherwise>
  </xsl:choose>
</xsl:template>
<xsl:template name="doSMTDescription">
  <!--
*****
|| Writes the appropriate SMT Description Section.
|| Input:
||   referenceDate: SMT (owning agency) reference date.
||
|| Example Call:
||   <xsl:variable name="SMTDescription">
||     <xsl:call-template name="doSMTDescription">
||       <xsl:with-param name="referenceDate" select="@j:ReportedDate"/>

```

```

||         <xsl:with-param name="agency" select="$SMTAgency"/>
||         <xsl:with-param name="codeValue" select="./scarsMarksTattoosDescription/@codeValue"/>
||         <xsl:with-param name="smtDescription" select="./scarsMarksTattoosDescription"/>
||         <xsl:with-param name="imgReferenceDate" select="./image/@j:ReportedDate"/>
||         <xsl:with-param name="imgAgency" select="$imgAgency"/>
||         <xsl:with-param name="imgHref" select="./image/imageData/@href"/>
||         <xsl:with-param name="imgComment" select="./image/comment"/>
||     </xsl:call-template>
*****

```

-->

```

<xsl:param name="referenceDate"/>
<xsl:param name="agency"/>
<xsl:param name="codeValue"/>
<xsl:param name="smtDescription"/>
<xsl:param name="imgReferenceDate"/>
<xsl:param name="imgAgency"/>
<xsl:param name="imgHref"/>
<xsl:param name="imgComment"/>
<!-- Put the SMT Description info together correctly -->
<xsl:if test="$codeValue">
    <xsl:value-of select="$codeValue"/>
    <xsl:if test="$smtDescription">
        <xsl:text>; </xsl:text>
    </xsl:if>
</xsl:if>
<xsl:if test="$smtDescription">
    <xsl:value-of select="$smtDescription"/>
</xsl:if>
<xsl:if test="$smtDescription or $codeValue">
    <xsl:text/>
</xsl:if>

```

```

<xsl:text></xsl:text>
<xsl:value-of select="$agency"/>
<!-- djr(19-MAR-03) agency is required -->
<xsl:if test="$referenceDate">
    <xsl:text>; </xsl:text>
    <xsl:value-of select="$referenceDate"/>
</xsl:if>
<xsl:text></xsl:text>
<xsl:if test="($smtDescription or $codeValue) and (string-length($imgComment)>0 or string-
length($imgReferenceDate)>0 or string-length($imgAgency)>0 or string-length($imgHref)>0)">
    <xsl:text/>
    <xsl:if test="$imgComment">
        <xsl:value-of select="$imgComment"/>
        <xsl:text/>
    </xsl:if>
    <xsl:if test="$imgHref">
        <xsl:value-of select="$imgHref"/>
        <xsl:text/>
    </xsl:if>
    <xsl:if test="$imgAgency">
        <xsl:text></xsl:text>
        <xsl:value-of select="$imgAgency"/>
        <xsl:text></xsl:text>
    </xsl:if>
</xsl:if>
</xsl:template>
<xsl:template name="doTypeImagesSection">
    <!--

```

\*\*\*\*\*

|| Writes the appropriate Type/Comments and Images Section. Used in the following Sections:  
|| -Fingerprint Images

```

|| -Palmprint Images
|| -Photo Images
||
|| Example Call:
||
*****

```

```

-->
  <xsl:param name="referenceDate"/>
  <xsl:param name="agency"/>
  <xsl:param name="Href"/>
  <xsl:param name="Comment"/>
  <!-- Put the info together correctly -->
  <xsl:if test="$Comment">
    <xsl:value-of select="$Comment"/>
    <xsl:text/>
  </xsl:if>
  <xsl:if test="$Href">
    <xsl:value-of select="$Href"/>
    <xsl:text/>
  </xsl:if>
  <xsl:text></xsl:text>
  <xsl:value-of select="$agency"/>
  <!-- djr(19-MAR-03) agency is required -->
  <xsl:if test="$referenceDate">
    <xsl:text>; </xsl:text>
    <xsl:value-of select="$referenceDate"/>
  </xsl:if>
  <xsl:text></xsl:text>
</xsl:template>

```

```

<xsl:template match="NewTemplate">

```

```
</xsl:template>  
  </xsl:stylesheet>
```



## Appendix D: Text

This appendix shows the result of transforming the example rap sheet of section 5 using the XSLT style sheet of Appendix C.

\*\*\*\*\* CRIMINAL HISTORY RECORD \*\*\*\*\*

Data As Of 2004-11-08

\*\*\*\*\* Introduction \*\*\*\*\*

This rap sheet was produced in response to the following request:

FBI Number 900660  
State Id Number CA99999978 (CA)  
Request Id 0HJGTM1THN  
Purpose Code C  
Attention AL 911137

The information in this rap sheet is subject to the following caveats:

This record is provided in response to your request. Use of the information contained in this record is governed by state and federal law. (CA; 2002-02-08)

\*\*\*\*\* IDENTIFICATION \*\*\*\*\*

Deceased 2002-05-12 (WI041015Y; 2001-02-27; Fingerprint Supported)

Subject Name(s)

Jones, MrJohn P Jr

Subject Description

FBI Number	State Id Number	DOC Number
62660CA12	CA9936278 (CA)	742786

Social Security Number	Driver's License Number
220565860	C4556289248R (WI)

Miscellaneous Numbers	State ID Card	WI
C4556289248R		

Sex	Race	Skin Tone
Female	White	Albino

Height                      Weight                      Date of Birth  
6'01" (1994-07-04)      180 (1994-07-04)      1953-02-12

Hair Color                  Eye Color                  Fingerprint Pattern  
Brown (1989-07-04)      Blue (1988-03-21)      66AA09TTPI58AA6413XI (FPC)

Scars, Marks, and Tattoos

Code                      Description, Comments, and Images  
TAT FARM                  tattoo, Dragon tattoo on right forearm.

Blood Type                  Medical Condition  
O Positive                  Suicidal (1964-04-01)  
Diabetic (1969-10-16)

Place of Birth              Citizenship              Ethnicity  
Virginia                  United States (2001-11-29)Hispanic Or Latino  
Unknown

Marital Status              Religion  
Never Married (2001-11-29)Lutheran (2001-11-29)

Employment

Employment as of      1987-09-15  
Occupation              Taxi Driver  
Employer                  Careful Cabs  
Location                  1565 N Park Place, Hamilton, NJ 08610

Residence

Residence as of      1987-09-15  
1565 N Park Place, Hamilton, NJ 08610

Residence as of      1989-04-22  
1565 N Park Place  
Apt B  
Hamilton, NJ 08610-1234

Telephone                  608-266-0872

Fingerprint Images

Fingerprint Image Available      CIB WI013415Y  
Available Image                  Right Index  
Capture Date                  1999-06-04  
Download URL

(Transmitted Image Suppressed:

Type:Fingerprint Format:image/wsqr Size:75K Comment:Grayscale)

Palmprint Images

Palmprint Image Available CIB WI013415Y  
Available Image Right Palm  
Capture Date 1999-06-04  
Download URL

(Transmitted Image Suppressed:

Type:Right Palm Format:image/wsqr Size:75K Comment:Grayscale)

Palmprint Image Available CIB WI013415Y

Available Image Both Palms  
Capture Date 1999-06-04

(No Palmprint Image Transmitted )

Photo Images

Photo Image Available CIB WI013415Y  
Available Image Identification  
Capture Date 1999-06-04  
Download URL

(Transmitted Image Suppressed:

Type:Identification Format:image/jpeg Size:15K Comment:Color pic)

Photo Image Available Wisconsin WI013415Y

Available Image Mugshot

(No Photo Image Transmitted )

DNA Data

DNA Information Available  
(No DNA Detail Transmitted )

DNA Sample Taken 2001-02-27

DNA Information Available

(Transmitted DNA Detail Suppressed ; Encoding MethodSTR  
Comment:Pursuant to sex offender registration.)

Caution Information

Firearms Disqualified true

Caution Armed and dangerous (CA; 1995-11-02)

Notice Subject of the record is a convicted felon.  
(CA; 1995-11-02)

\*\*\*\*\* CRIMINAL HISTORY \*\*\*\*\*

==== Cycle 001 =====  
Earliest Event Date 1995-07-04 Incident Date 1998-05-28

-----  
Arrest Date 1998-05-30  
Arrest Case Number 1998AF002354  
Arresting Agency WI013415Y

Subject's Name Mitch Doherty  
Michael Doherty  
Offender Id Number 48204395  
Arrest Type Adult  
Comment(s) Resisted arrest  
Charge 01  
Charge Number 94D002356  
Charge Tracking Number AD486  
Charge Literal Robbery  
Charge Description Referred for clinical evaluation  
Statute Statute Text (943.2(e)10; WI)  
NCIC Offense Code 0106  
State Offense Code 20836  
Counts 4  
Severity Felony  
Inchoate Charge Attempt  
Enhancing Factor Elderly victim  
Reducing Factor First offense  
Disposition (Charges Dropped 1996-07-07; Case closed  
without referral for prosecution.)

-----  
Booking Case Number 1998AF002355  
Booking Agency CIB WI013415Y  
-----

Prosecutor Disposition (Cycle 001)  
Prosecutor Case Number 94D002356  
Prosecution Date 1998-05-30  
Prosecutor Agency WI013415Y CIB  
Subject's Name Mitch Doherty  
Michael Doherty  
Charge 01  
Charge Number 94D002356  
Charge Tracking Number AD486  
Charge Literal Burglary  
Charge Description Referred for clinical evaluation  
Statute Text (943.2(e)10; WI)  
NCIC Offense Code 0106  
State Offense Code 20836  
Counts 4  
Severity Felony  
Inchoate Charge Attempt  
Enhancing Factor Elderly victim  
Reducing Factor First offense  
Disposition (Not Prosecuted 1996-07-07; Case closed without  
court filing.)  
-----

Court Disposition (Cycle 001)  
Court Case Number 1998AF002356  
Court Agency WI014400Y  
Subject's Name Mitch Doherty  
Michael Doherty  
Charge 01  
Charge Number 94D002356  
Charge Tracking Number AD486  
Charge Literal Trespassing  
Charge Description Referred for clinical evaluation  
Statute Text (943.2(e)10; WI)  
NCIC Offense Code 0106  
State Offense Code 20836  
Counts 4  
Severity Felony  
Inchoate Charge Attempt  
Enhancing Factor Elderly victim  
Reducing Factor First offense  
Disposition (Convicted 1996-07-07; Guilty.)

---

Sentencing (Cycle 001)  
Sentence Date 1998-05-30  
Sentencing Agency Portage County Circuit Court  
Court Case Number 1998AF002356  
Charge 01  
Charge Number 94D002356  
Charge Tracking Number AD486  
Charge Literal Trespassing  
Sentence 3Y Prison  
Sentence \$500 fine and costs

---

Corrections (Cycle 001)  
Corrections Agency Portage County Jail  
Subject's Name Mitch Doherty  
Michael Doherty

Correctional Id Number 48204395  
Correction Action incarcerated  
Correction Comment Probation revoked.  
Release Date 1998-06-30

\*\*\*\*\* INDEX OF AGENCIES \*\*\*\*\*

Agency Lodi Police Department; WI0111000;  
Contact Records Department  
George  
Agency Telephone 4859304869  
Agency Facsimile 4859304870

Agency Email Address george@lodiPD.gov  
Address

1565 N Park Place

\* \* \* END OF RECORD \* \* \*

## *Appendix F: Reference Documents*

The following documents provide source material upon which this document was based.

1. *Interstate Criminal History Transmission Specification*, Version 2.22, August 2003, Joint Task Force on Rap Sheet Standardization.
2. *Global Justice XML Data Model*, Version 3.0, 2004, Georgia Technical Research Institute
3. *Data Format for the Interchange of Fingerprint Information*, American National Standards Institute and National Institute for Standards and Technology, ANSI/NIST-ITL -2000.  
<http://www.itl.nist.gov/iad/894.03>
4. *Increasing the Utility of the Criminal History Record: Report of the National Task Force*, Bureau of Justice Statistics, U. S. Department of Justice, NCJ-156922, December 1995.
5. Interstate Identification Index (III) Operational and Technical Manual, August 1994.
6. Extensible Markup Language (XML) 1.0 Specification, <http://www.w3.org/TR/REC-xml>, including errata at <http://www.w3.org/XML/xml-19980210-errata>.
7. Extensible Stylesheet Language (XSL), <http://www.w3.org/TR/xsl>.
8. Extensible Stylesheet Language: Transformations (XSLT), <http://www.w3.org/TR/xslt>.
9. NIST: American National Standard for Information Systems – Data Format for the Interchange of Fingerprint Information, ANSI/NIST-CSL-1-1993.<sup>9</sup>
10. Task Force Report: *Increasing the Utility of the Criminal History Record: Report of the National Task Force*, Bureau of Justice Statistics, U.S. Department of Justice, NCJ-156922, December 1995.
11. Interstate Identification Index (III) Operational and Technical Manual, August 1994.
12. EFTS: Electronic Fingerprint Transmission Specification, CJIS-RS-0010, FBI CJIS Division.
13. National Crime Information Center 2000 Code Manual

The following documents contain background and tutorial material.

1. *Professional XML*, D. Martin, M. Birbeck, M. Kay, B Loesgren, J. Pinnock, S. Livingstone, P. Stark, K. Williams, R. Anderson, S. Mohr, D. Baliles, B. Peat, N. Ozu.

---

<sup>9</sup>ANSI/NIST = American National Standards Institute/National Institute of Standards and Technology.

2. *XSLT Programmer's Reference*, Michael Kay, Wrox Press, Ltd., 2000.
3. *XSL Concepts and Practical Use*, Paul Grosso and Norman Walsh, <http://www.nwalsh.com/docs/tutorials/xsl/xsl/slides.html>.
4. *XML Namespaces*, <http://www.w3.org/TR/1999/REC-xml-names-19990114>.
5. *XML Namespaces*, James Clark, <http://www.jclark.com/xml/xmlns.htm>
6. *XML Schema Part 0: Primer*, W3C Working Draft, <http://www.w3.org/TR/2000/WD-xmlschema-0-20000407>.
7. *XML Schema Part 1: Structures* W3C Working Draft, <http://www.w3.org/TR/2000/WD-xmlschema-1-20000407>.
8. *XML Schema Part 2: Datatypes*, W3C Working Draft, <http://www.w3.org/TR/2000/WD-xmlschema-2-2000040>



# *Appendix G: Joint Task Force on Rap Sheet*

## **Standardization Participants**

**CHAIRMAN:**            **Gerry Coleman**  
Wisconsin Department of Justice

**EX-CHAIRMAN:**        **John Loverude**  
Illinois State Police (former)  
Advanced Technology Systems, Inc. (current)

**CURRENT MEMEBERS:** **Owen M. Greenspan**  
SEARCH, the National Consortium for Justice Information and Statistics

**Mike Lesko**  
Texas Department of Public Safety

**Tom Hopper**  
CJIS Division, Federal Bureau of Investigation

**Steve Correll**  
National Law Enforcement Telecommunications System

**Charlie Pruitt**  
Arkansas Crime Information Center

**PARTICIPANTS:**        **William Casey\***  
Boston Police Department

**David Gavin\*\***  
Administration Division, Texas Department of Public Safety

**Debra M. Jenkins\*\*\***  
U.S. Marshals Service

**Jim Martin+**  
South Carolina Law Enforcement Division

**Robert L. Marx+**  
SEARCH, the National Consortium for Justice Information and Statistics

**Tim Sweeney+**  
National Law Enforcement Telecommunications System

**Dr. Paul Anderson**  
PNL Associates, LLC

**Frank Minice**  
National Law Enforcement Telecommunications System

**Bob Slaski**  
Advanced Technology Systems, Inc.

**FBI CJIS SUPPORT:**    **James Gerst**  
                                  **Gary Barron**  
                                  **Andy Herberger**  
                                  **Dean Manson**  
                                  **Lottie Martin**

+ Indicates past JTF member

- \* Bill Casey served on the JTF in his capacity as 1st Vice Chair of the FBI CJIS Advisory Policy Board (APB)
- \*\* David Gavin served on the JTF in his capacity as Chair of the APB
- \*\*\* Debra Jenkins served as a representative of the FBI CJIS Federal Working Group

# *Appendix H: National Task Force on Increasing the Utility of the Criminal History Record (1993-1995)*

- Chairman:** **Jack Scheidegger**  
Chief, Bureau of Criminal Identification and Information, California Department of Justice
- Participants:** **Kenneth E. Bischoff**  
Director, Administrative Services, Alaska Department of Public Safety
- Joseph P. Bonino**  
Chairman, Criminal Justice Information Services Advisory Policy Board: Commanding Officer, Records and Identification Division, Los Angeles, California Police Department
- Lt. Larry Copley**  
Commanding Officer, Identification Section, Central Records Division, Michigan Department of State Police
- Patrick J. Doyle**  
Immediate Past Chairman, National Crime Information Center Advisory Policy Board: Director, Division of Criminal Justice Information Systems, Florida Department of Law Enforcement
- Owen M. Greenspan**  
Deputy Commissioner, New York State Division of Criminal Justice Services
- Bruce M. Harvey**  
Milwaukee County, Wisconsin Justice System Coordinator
- Dr. Sally Hillsman**  
Vice President, Research and Technical Services, National Center for State Courts
- Robert R. Hole**  
Deputy District Attorney, Contra Costa County, California
- Honorable Michael Hutchings**  
Utah Third Circuit Court
- Frank Johnstone**  
Section Chief, Technical Services Section, Information Systems and Technology Division, Virginia Department of Criminal Justice Services
- Carol G. Kaplan**  
Project Monitor, Assistant Deputy Director, Bureau of Justice Statistics, U.S. Department of Justice
- George Klier**  
Bureau Chief, Information Services, Norfolk County, Massachusetts Sheriff's Office
- David B. Lodge**  
Special Agent-in-Charge, Crystal City Investigative Field Office, Defense Investigative Service, U.S. Department of Defense
- Edward J. Loughran**  
Director, Juvenile Justice Project, Robert F. Kennedy Memorial

**Clifford H. Marshall**  
Sheriff, Norfolk County, Massachusetts

**Jerome E. McElroy**  
Director, New York City Criminal Justice Agency

**Judy Metz**  
Chief, Correctional Case Records Services, California Department of Corrections

**Matthew Myers**  
Undersheriff, Ingham County, Michigan

**Rosemarie Pifer**  
Director, Central Records Division, Michigan Department of State Police

**Emmet A. Rathbun**  
Unit Chief, Criminal Justice Information Services Division, Federal Bureau of Investigation

**Anthony L. Stolz Jr.**  
Director, Personnel Investigations Center, Defense Investigative Service, U.S. Department of Defense

**Capt. R. Lewis Vass**  
Records Management Officer, Records Management Division, Virginia State Police

**Lawrence P. Webster**  
Director, Court Technology Programs, National Center for State Courts

**Gene Wriggelsworth**  
Sheriff, Ingham County Michigan

**Gary T. Yancy**  
District Attorney, Contra Costa County, California

**Virgil L. Young Jr.**  
Section Chief, Programs Development Section, Criminal Justice Information Services Division, Federal Bureau of Investigation