

A Prototype for XML-based E-Filing of Criminal Complaints

Using the JXDD 3.0, ebXML Messaging Handling Service 2.0, and a customized LegalXML Court Filing 1.1 based schema to implement a prototype e-filing delivery system for criminal complaint documents.

October, 2003

A report on research sponsored by the Los Angeles County Information Systems Advisory Body and conducted by Sierra Systems.

<http://www.sierrasystems.com>



Introduction.....	4
Reference XML Standards for Project Development	4
Our Approach to Mapping Criminal Complaints and Juvenile Petitions to XML	6
Definitions.....	6
What did we do (and why)?	6
Specific Modifications	9
Problems Encountered and Recommendations.....	11
Conclusions.....	11
Prototype Implementation.....	12
What were the functional requirements for the prototype?	12
What ebXML compliant products were considered?.....	13
What product did we choose and why?	14
Prototype Architecture	15
The Prototype in Action.....	19
Lessons Learned.....	19
Performance Testing the Prototype.....	20
Results.....	23
Final Notes	24
Summary.....	24
Credits.....	25
Appendix A - Resources	26
Appendix B - Sample XML Filing Response	27
Appendix C - Sample XML Filing (Felony Complaint w/o Attachments)	28
Appendix D - LA County Sample Felony Complaint (BLOB)	33

Introduction

The Los Angeles County Information Systems Advisory Body (ISAB) sponsored the development of a secured delivery method for filing prosecution documents, specifically the *criminal complaint*, with the County of Los Angeles Superior Court. Currently, the LA County District Attorney Office performs e-filing of case management data from the Prosecutor Information Management System (PIMS) but hand carries the complaint document for the court clerk to file. This pilot project will enhance the e-filing process to include the complaint document and PIMS case filing data in a single e-filing process utilizing evolving Justice XML standards and an existing secured message delivery standard. The message delivery standard is referred to as Message Handling Services for Electronic Business XML (MSH v2.x for ebXML).

The efforts of this project were divided between two goals: to create a prototype set of XML-mapped criminal filings that rely on the Justice XML Data Dictionary (JXDD) and take structural cues from the LegalXML Court Filing 1.1 (CF1.1) standard, and to implement a prototype e-filing delivery system using the ebXML Messaging Handling Services.

By moving XML complaint filings through a prototype delivery system, we aimed to assess the strengths and weaknesses of ebXML as a delivery mechanism, and to find missing or misplaced components of the JXDD and CF1.1 that affect the handling of complaint filings. From this process we derived recommendations regarding each of the technologies and standards used.

Reference XML Standards for Project Development

LegalXML Court Filing 1.1

LegalXML is a member section of the non-profit OASIS consortium that “brings legal and technical experts together to create standards for the electronic exchange of legal data.”¹ Court Filing 1.1 is an early-proposed standard from the LegalXML technical committee on court filing.

Court Filing provides a single DTD for validating XML formatted court filings. The DTD includes document structure for filings, responses to filings, queries, and responses to queries; as well an XML vocabulary for court related subjects. The CF1.1 proposed standard also specifies a high-level architecture and XML protocols for sending and receiving court filing documents.

JXDD 3.0

A subset of the Justice XML Project, the JXDD provides “An object-oriented data model, database, and XML schema specification (generated from the database) that represent the

¹ <http://www.legalxml.org/about/index.shtml>

semantics and structure of common data elements and types required to exchange information consistently within the justice and public safety communities.²”

The JXDD is currently dependant on approximately 20 external schemas, and internally describes approximately 2,000 components relating directly to justice and public safety. The JXDD is defined using XML schema, a newer standard that provides for more flexibility than DTD, which Court Filing 1.1 uses.

ebXML Messaging Service 2.0

“ebXML (Electronic Business using eXtensible Markup Language), sponsored by UN/CEFACT and OASIS, is a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet.³” The ebXML Message Specification 2.0 (ebMS) defines a protocol for businesses to communicate transaction information using XML over the Internet. A number of companies provide ebMS software that function as “Message Service Handlers.” These MSH packages generally provide installable software that enable computers to send and receive ebMS messages securely and reliably.

² <http://it.ojp.gov/jxdd/faq.html>, <http://justicexml.gtri.gatech.edu/>

³ <http://www.ebxml.org/geninfo.htm>

Our Approach to Mapping Criminal Complaints and Juvenile Petitions to XML

Definitions

The following definitions apply within this section:

- **CF1.1** refers to the Court Filing 1.1 standard prior to our customization
- ***cf11:elementName*** refers to an element in the Court Filing 1.1 schema namespace
- ***jxdd: elementName*** refers to an element in the JXDD 3.0 schema namespace
- ***ebms:elementName*** refers to an element in the ebXML Messaging Service 2.0 namespace.

What did we do (and why)?

The court-filing prototype examined new possibilities for the creation and filing of criminal complaints that would best leverage emergent standards and technologies, as well as solve domain specific problems.

A set of LA County District Attorney complaint filings (1 juvenile petition and 1 criminal complaint) were mapped, using a customized schema that borrows from the CF1.1 DTD but uses JXDD 3.0 data elements and defers messaging elements to the ebXML messaging infrastructure. In doing so, we first ran an automated conversion of the CF1.1 DTD to an XML schema (using Trang⁴) in order to make it easier to integrate with the large number of schemas linked into the JXDD. We then customized the CF1.1 schema using the following guidelines:

1. For our purposes, ebXML provides a wire (transmission) protocol.
2. For our purposes, CF1.1 provides a document management framework.
3. For our purposes, the JXDD provides consistently defined data description elements within the justice domain.
4. We replaced CF1.1 elements with JXDD wherever the ONLY function of the CF1.1 is to describe data.
5. We use CF1.1 elements when they provided document(s) with cardinal structure.
6. We use CF1.1 elements when they provided document version information.
7. We retained the ebXML messaging software without modification.

⁴ <http://www.thaiopensource.com/relaxng/trang.html>

8. We retained the CF1.1 *legalEnvelope* element as the root XML element for assembling our filings.
9. The actual court documents were included as “blobs”⁵ within the Court Filing XML document

In general, modifications to the Court Filing 1.1 schema were restricted to areas relating to *filing* and *filing confirmation*. Our prototype system doesn’t currently support concepts of *query* and *response* and those elements remain unchanged. Other non-compliant related elements of CF1.1 were also considered beyond the scope of this exercise.

The following sections describe significant issues discovered in creating the custom CF1.1 based schema.

Cardinality and the JXDD

One significant issue in our work has been at what depth in our CF1.1 based schema to incorporate JXDD elements. Primarily, our concern is with min/max occurrence within the custom CF1.1 schema. Being a dictionary only, the JXDD has no rules on cardinality, so any JXDD element introduced into our CF1.1 based schema makes it possible to introduce an “empty” element where the CF1.1 based schema would have required a minimal or maximal number of occurrences of a particular element.

We generally choose to retain CF1.1 elements in any case where the element in question provides a clear function in structuring the filing document. For example, we retain the CF1.1 elements for *filingInformation* and *legalEnvelope* because they enforce the minimally necessary items that should be included in a proper filing.

Senders and Addressees

CF1.1 contains its own mechanism for routing messages to recipients and providing recipients with a channel to respond to those messages. ebXML implements sender and recipient information by placing references in the ebXML envelope that provide enough information to deliver messages to a precise address, but also may be linked to more comprehensive information about the parties involved in the transaction in a separate ebXML document called a Collaboration Protocol Profile (CPP⁶). We chose to use ebXML for these functions as opposed to the CF1.1 legal envelope elements. Our prototype has removed the *cf11:to*, *cf11:from*, *cf11:cc*, and *cf11:bcc* elements from the *cf11:legalEnvelope*.

⁵ Binary Large Objects.

⁶ The CPP is part of the Registry and Repository service in ebXML that we did not implement for our prototype.

Actors

A *cf11:actor* may have a “role” that establishes a relationship between one *cf11:actor* and either another *cf11:actor* or a *cf11:matter*. The JXDD supports an idea of roles, however it establishes them quite differently. The JXDD uses an object-based model- so that there may be sub-classes of *jxdd:PersonType* or *jxdd:ActorType* whose roles are implied by their name, e.g. *jxdd:CaseDefendantActor* or *jxdd:CaseDefenseAttorney*. In the case of *jxdd:CaseOfficialType* elements, a *jxdd:CaseOfficialRoleText* element is provided that is similar to the *cf11:role* element.

The difference between *cf11:actor* and *jxdd:Actor* presented a number of alternatives for this project.

- 1.) We could retain *cf11:actor* as it exists (DTD shown⁷):

```
<!ELEMENT actor (title?, name?, postalAddress*, telephone*,
                 email*, group*, personDescription*,
                 designation*, role*, characteristic*)>
```

- 2.) We could retain the *cf11:actor* element, but replace elements within *cf11:actor* with their JXDD counterparts:

```
<!ELEMENT actor (JXDD:PersonPrefixName8? JXDD:PersonName?,
                 JXDD:Location*, JXDD:ContactInformation*...>
```

- 3.) We could replace *cf11:actor* entirely with *jxdd:Actor* (or- more appropriate to our scenario, we could replace them with *jxdd:CaseParticipants*):

```
<!ELEMENT filingInformation (specialHandling?,...actor*...)>
```

becomes:

```
<!ELEMENT filingInformation (specialHandling?,...
                             JXDD:CaseParticipants...)>
```

In the end we chose option 3, to replace lists of *cf11:actors* with *jxdd:CaseParticipants*, an element that holds an arbitrary number of *jxdd:Actor* elements and elements that extend from the *jxdd:Actor* such as *jxdd:CaseDefendantActor* and *jxdd:CaseInitiatingActor*. *jxdd:Actor* is significantly more expansive as a data type than *cf11:actor*. More importantly, moving to the JXDD model provides the ability to

⁷ We have chosen to use DTD in our examples so that users can easily find refer to the original Court Filing 1.1 specification. Also, the DTD code snippets are exceptional more compact than the equivalent XML Schema.

⁸ This is not true DTD, as DTD does not support multiple namespaces. Assume that, in our custom schema, elements prefixed by “jxdd:” refer to JXDD elements while non-prefixed elements refer to Court Filing 1.1 elements.

guarantee normalized comparison between other JXDD entities. The downside to this is that we lost the ability to use *cf11:roles* in some cases.

Dispositions

CF1.1 provides a mechanism for describing the disposition of cases, documents, and supporting documents. JXDD has similar elements. The fundamental difference between the two specifications is that the disposition elements within the CF1.1 spec provide strict controls over the options that a disposition may carry. For instance, a *cf11:filingDisposition* may ONLY carry the value of *acknowledged* or *transmissionError*. JXDD disposition elements on the other-hand simply provide a string value representing whatever the user desires. We have chosen to leave all of the CF1.1 disposition elements as is because the specified disposition responses relate directly to our domain specific needs.

Specific Modifications

The following specific modifications were made to our custom CF1.1 based schema. For simplicity, examples shown are DTD⁹ code, not the actual code from our custom schema.

cf11:to, *cf11:from*, *cf11:cc*, *cf11:bcc*, *cf:replyTo*, *cf11:dataIntegrity*, *cf11:authentication* were removed from the *cf11:legalEnvelope* element because ebXML provides these functions.

Before:

```
<!ELEMENT legalEnvelope (messageIdentification, to, from, cc?, bcc?,  
replyTo?, memo*, creation, dataIntegrity?,  
paymentInformation? authentication?, legal)>
```

After:

```
<!ELEMENT legalEnvelope (messageIdentification, memo*, creation,  
paymentInformation? legal)>
```

⁹ This isn't true DTD, as DTD doesn't support multiple namespaces. Assume elements prefixed by "jxdd:" refer to JXDD elements while non-prefixed elements refer to Court Filing 1.1 elements in our custom schema.

cf11:filingInformation and ***cf11:confirmationInformation*** were revised to use ***jxdd:Case***. ***jxdd:Case*** internally includes ***jxdd:CaseParticipants***, ***jxdd:Court***, and ***jxdd:CaseCourtEvent***.

Before:

```
<!ELEMENT filingInformation (specialHandling?, paymentInformation*,  
                             authentication*, courtInformation,  
                             caseInformation, actor*, courtEvent*,  
                             memo?)>
```

After:

```
<!ELEMENT filingInformation (specialHandling?, paymentInformation*,  
                             jxdd:Court, jxdd:Case, memo?)>
```

cf11:documentInformation and ***cf11:attachmentDocumentInformation*** were revised to use ***jxdd:CaseParticipants***, ***jxdd:Submission***, and ***jxdd:Document*** in place of ***cf11:actor***, ***cf11:submitted***, and ***cf11:documentDescription***

Before:

```
<!ELEMENT documentInformation (actor+, submitted, documentDescription,  
                               (administrativeLaw | appeals |  
                               bankruptcy | civil | criminal |  
                               domesticRelations | juvenile | probate |  
                               smallClaims | traffic)?, matter*,  
                               causeOfAction*)>
```

After:

```
<!ELEMENT documentInformation (jxdd:CaseParticipants, jxdd:Submission,  
                               jxdd:Document, (administrativeLaw |  
                               appeals | bankruptcy | civil | criminal  
                               | domesticRelations | juvenile | probate  
                               | smallClaims | traffic)?, matter*,  
                               causeOfAction*)>
```

<p><i>cf11:criminal</i> and <i>cf11:juvenile</i> were modified to use <i>jxdd:Actor</i>, <i>jxdd:Arrest</i>, <i>jxdd:Booking</i>, and <i>jxdd:Offense</i>.</p>
<p>Before:</p> <pre><!ELEMENT criminal (identification, charge+, bail?, booking?, custody?, incident*)></pre>
<p>After:</p> <pre><!ELEMENT criminal (jxdd:CaseParticipants, jxdd:Charge+, jxdd:Arrest?, jxdd:Booking?, jxdd:Offense*)></pre>

Problems Encountered and Recommendations

We had a handful of unresolved issues in our mapping mostly within the JXDD.

- JXDD secondary relationships proved too cumbersome for our use.
- There is no way to specify the last school a person attended, only the highest grade they attended using *jxdd:PersonEducationLevelText*.

Conclusions

Using the custom CF1.1 based schema with the modifications detailed above, we were able to map both a felony complaint (see Appendix C) and a juvenile petition. We found that our solution was generally successful, with the few exceptions of the elements described in the preceding section.

Prototype Implementation

What were the functional requirements for the prototype?

ebXML was designed to solve a number of problems in ways that match e-filing business needs. These include securing the identification of parties involved in the exchange of documents, securing the content while in transit, and guaranteeing that documents are delivered reliably. ebXML supports the development of low-cost and highly interoperable software by implementing its Messaging Service protocol. ebXML also specifies a number of collaboration protocol services that we did not currently require, such as the registration of exchange capabilities and services within the ebXML registry. Fortunately, ebXML is a modular set of specifications and no additional work was required to exclusively use the ebXML messaging services functions.

Security

Our prototype system is designed to exchange sensitive information across the WAN or Internet. It is crucial that senders and receivers are able to authenticate that they are who they say they are, and that no one can read or modify documents that aren't intended for them. For our prototype we searched for an ebXML product that would enable us to use SSL encrypted HTTPS connections with certificate based authentication.

Reliability

It is important for our prototype that documents are sent reliably- meaning that when a connection is broken mid-transfer, both the sender and receiver are aware of that break and how to properly deal with it.

Cost

As a prototype, we chose to make a minimal investment in software.

Platform considerations

Our prototype must be capable of running on a number of platforms- specifically, HP/UX, Windows 2000, Windows XP, and possibly LINUX.

Why we didn't utilize ebXML registry services?

ebXML provides a layer for "discovering" services. In practice, two court systems from two different states could use such a service to find each other and learn *how* to communicate documents between each other. ebXML's Registry and Repository service as well as UDDI¹⁰ also exist for this purpose. While either of these might add functionality to our prototype, we have chosen to avoid using any kind of registry, assuming that for our purposes criminal filing(s) will be between known sender and receiver organizations.

¹⁰ Universal Description, Discovery, and Integration: <http://www.uddi.org/>

What ebXML compliant products were considered?

Options

A fairly substantial number of ebXML MSH software packages are available, both in commercial and open source formats. Of the existing options, we examined the following three packages: CEBIT Hermes (available through FreebXML.org), Sybase's Open Source ebXML MSH, and Sun's Secure Transaction Server 1.0. The first two of these packages are available freely through open source licenses. The last is a commercial product.

FreebXML - <http://www.freebxml.org/msh.htm>

Hermes is produced by the Center for E-Commerce Infrastructure Development at the University of Hong Kong and distributed by the FreebXML initiative. CECID is a member of the OASIS group that oversees XML projects, in particular ebXML. FreebXML is a centralized site for developers to share and access "free" ebXML code and applications. Hermes is provided under the Academic Free License¹¹.

FreebXML
Strengths
Free
Has been implemented for similar scaled projects
Passed Drummond Asian interoperability testing
Supports QOS
Platform independent
Weakness
Requires developer to setup and optimize servlet container separately

¹¹ <http://www.opensource.org/licenses/academic.php>

Sybase Open Source ebXML Messaging - <http://www.sybase.com/developer/opensource>

Sybase Open Source ebXML is provided via the Sybase Open Source License and sourced from the software they provide commercially through their Web Integration Services product.

Sybase Open Source ebXML Messaging

Strengths

Free

Passed Drummond group interoperability in March 2002

Based on Sybase's enterprise product for ebXML messaging

Can be implemented through both Servlets and J2EE

Weakness

Fair number of API dependencies (5-10), some of which are deprecated

Requires developer to setup and optimize servlet container separately

Sun ONE Secure Trading Agent 1.0 -

http://www.sun.com/software/products/integration_srvr_sta/

Sun is one of the principal forces behind ebXML. Secure Trading Agent is Sun's implementation of the MSH protocol and operates as part of the Sun ONE Integration Server. STA installs and configures its own instance of Tomcat server.

Sun ONE Secure Trading Agent 1.0

Strengths

Easy to setup

Weakness

Not free (commercial release)

Only runs on Solaris and Windows

What product did we choose and why?

We chose to implement our prototype using CECID's Hermes MSH v.9.3.1. The following issues informed our decision:

- 1.) The software is licensed for free.
- 2.) It is open source. We needed to do a lot of API level work, and for a relatively emergent product, to be able to look inside when things didn't work as advertised was a benefit.
- 3.) Hermes is written in JAVA, and comes with all of the benefits contained therein, most importantly platform independence and ease of coding.

4.) Hermes has been recently implemented on similar projects.

Overall, Hermes has proven a good choice for us. Despite being a non-commercial product, online support through members of the development team and the user community is strong. Installation is easy enough (we now routinely install the software from scratch in under 15 minutes) and reliability has been within our tolerance. The product underwent a point upgrade while we were using it that corrected the few bugs we were experiencing.

Prototype Architecture

CF1.1 specified “EFM”

Our prototype is based on the architecture specified by CF1.1. Under that specification, three software components work together to complete filings. A user who wants to submit a file utilizes an Electronic Filing Provider (EFP). This is essentially a “client.” The EFP sends the file to an Electronic Filing Manger (EFM), the *server* software that receives the file. The EFP in turn hands the successfully delivered filing to a Case Management System (CMS). Within the scope of our prototype, we only focused on the interaction of the EFP and the EFM. Our MSH, Hermes, provides a usable graphically driven client application for sending files to any ebMS MSH that we utilize as an EFP. Our only requirement therefore was to write software for the EFM.

The specific jobs of an EFM are to properly handle the secure receipt of filings and to acknowledge/confirm receipt back to the sender, either manually or automatically. ebXML as the messaging service provides secure and reliable messaging. Additionally, the e-filing content must be validated before the EFM can “accept” or “reject” them. The EFM prototype validates the XML against the custom CF1.1 schema. In a real world implementation, we would also perform virus scanning before accepting the filing. While we did not code automatic virus scanning capabilities into our prototype, we did perform virus scanning performance tests of the complaint filings.

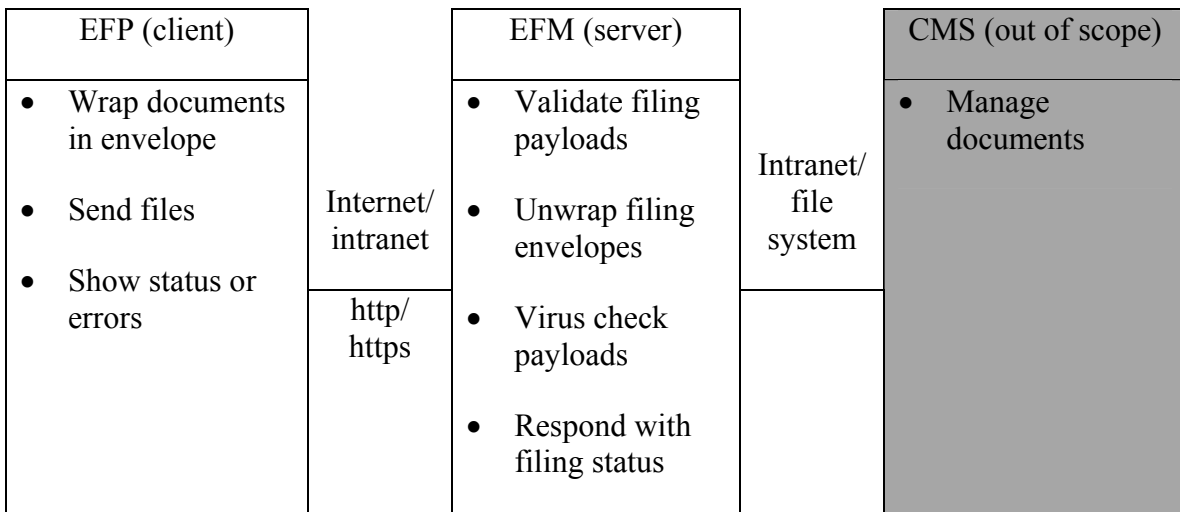


Diagram – Components of our prototype architecture.

What EFP/EFM components we did not build

Hermes provides a graphical client for sending ebXML payloads using any MSH with any number of ebMS functions enabled. This includes reliable messaging, digital signatures, and duplicate message elimination. Because ebMS is a peer-to-peer system, the sending client also has receiving functions. When our EFM returns “received”, “accepted”, or “declined” messages, the Hermes client can display those XML response messages along with any related documents that may be attached.

What the Hermes client does not provide is a means of constructing the custom CF1.1 compliant XML and content. Instead of creating software to perform this “wrapping”, we created our filings manually, before submitting them as attachments to ebXML messages. Additionally, ebXML MSH supports digital signature for the “payloads” (BLOBS) which we did not prototype for this project.

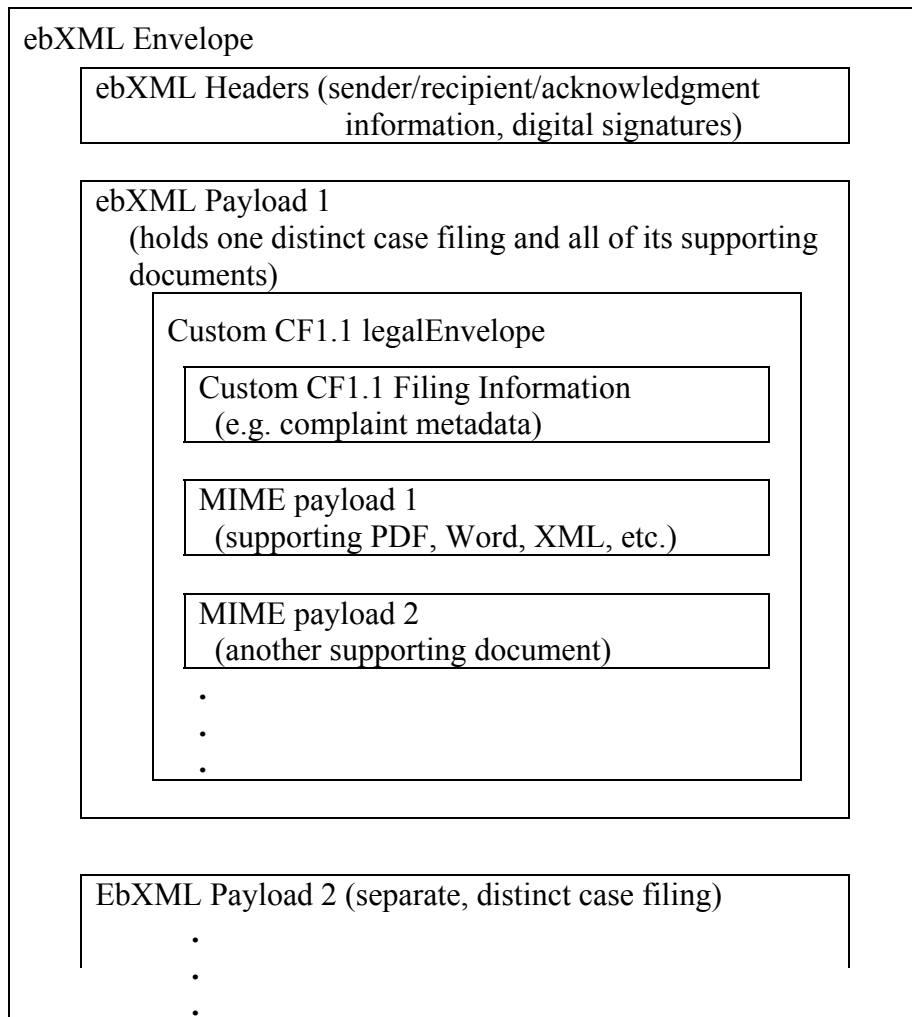


Diagram: how the ebXML envelope “wraps” around custom CF1.1 envelope and payload documents.

What EFP/EFM components we did build

Our EFM utilizes Herme’s open API. Using it, we created software for receiving ebMS messages, retrieving the custom CF1.1 filing within the ebXML messages, validating the filing against the custom CF1.1 schema, and then automatically routing the filing into either a “rejected” or “received” directory based the success or failure of the XML filing schema validation. The two directories are accessible to end-users by simply browsing through the file system. For the purpose of prototype auditing and testing, end-users can review the “received” and “rejected” filings manually and then drag and drop the files into a “received” directory, a “rejected” directory, or an “accepted” directory. The appearance of any new files in the “accepted”, “rejected”, or “received” directories triggers a file monitor program and causes a message to be automatically sent back to the sender with a unique message ID to correlate the response to the original filing, and a notice of the updated status of the filing. The “accepted” status is reserved for the second stage of filing where a clerk may review the filing, or a CMS may automate the filing and then a return status of “accepted”, “rejected” etc. would come from the CMS back to the EFM and optionally back to the original filer depending on Court policy and agreements between filers and the EFM.

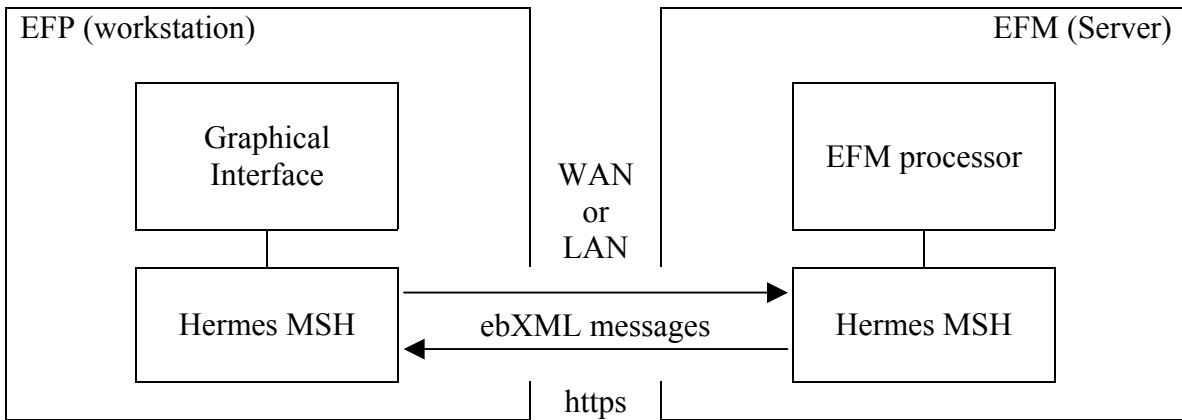


Diagram: EFP and EFM

Prototype Design Details

Schema caching

Our initial concern regarding the JXDD is the length of time required to validate JXDD compliant XML files. Knowing that we would have to validate every incoming message we received, we adopted an efficient means for validation. Our prototype uses the Xerces SAX parser. Xerces caches XML grammars as they load; resulting in decreased validation times for some of our performance tests by as much as 75%.

Response generation

Court Filing 1.1 specifies domain specific message responses as well transmission reliability responses, while ebXML only contains the latter. All CF1.1 messages contain a *messageIdentification* element and a *creation* element, which together define a specific message in a related series of messages. Recipient machines use these elements to provide error notification as well as workflow status messages. Status messages such as *transmissionFailed*, *received*, *rejected*, *accepted*, and *deferred* are all defined within the CF1.1.

We rely on ebXML and its internal messageIDs for reliable messaging transmission responses. Our review concluded that the message delivery responses of ebXML provided a robust process for ensuring reliable message delivery (replacing *cf11:transmissionFailed,received*) and we implemented a second layer of response messages based on the CF1.1 messageIdentification and response codes to provide status updates that need to be addressed by filers, EFMs and Court personnel (*cf11:rejected,accepted,deferred*). A good example of the logical separation of delivery statuses from filing content statuses would be using express mail to deliver a number of court documents. The express mail envelope would likely have a tracking number for locating the documents while in transit, but each document would contain a court specific identifier for tracking it within the court system. Any response to a court document would reference the court identification number, not the express mail identifier, as the two processes are primarily independent. See appendix A for a sample response message.

Implications of design

Because our system is only a prototype, we did not produce it to be end-user oriented, nor bulletproof in its handling of unexpected document types. Our simple goals were to quickly produce a system that would enable a number of critical tests regarding the usability of ebXML as a messaging protocol for our filing documents. While no work was invested in providing a robust user interface or exceptional error handling of rare error scenarios, this system does implement ebXML to the current specifications of its security and reliable messaging capabilities. We can reasonably assume that our test results would not be strongly affected by having a sophisticated user interface, and that in a real-world scenario, a true EFP would act as a gatekeeper on the types of documents that would be successfully transmitted to an EFM.

The Prototype in Action

Lessons Learned

Ease of installation

Hermes is installed as a set of JAVA servlets launched within any Sun servlet container. For our installation, we chose to use the Apache Group's Tomcat¹² (v.4.1.27) servlet container, which is Sun's reference implementation of the servlet specification. The Hermes installation provides a pre-built WAR archive that is simply copied into Tomcat's deployment directory, causing it to install itself and all third party libraries that Hermes depends on.

After this, the installation became more difficult. Hermes depends on a number of setup files that are very sensitive to network configuration. This is especially the case when certificate security is used for SSL connections between senders and receivers. Hermes validates the machines involved in exchanges by storing the fully qualified domain name of each machine in its certificate. As a result, changing the name of a machine will invalidate its certificate for use as a Hermes MSH. This also means that certificates cannot be moved easily from one machine to another unless the target machine takes the network place of the original machine. Despite these issues, once configured the setup was consistently reliable.

Operation

We wrote a testing harness that composed and sent ebMS messages from one MSH to another with any number of attachments and a user definable rate of repetition. The harness was written using Hermes API. We can run the testing harness from any MSH to any other MSH on our network. Moreover, we can run the testing harness concurrently on multiple machines on our network to simulate real-world network loads.

Network Architecture

We have three machines networked for testing purposes. One HP-UX machine and two Windows 2000 machines are placed on the WAN. The two W2K machines on the WAN are within the same domain, while the HP-UX machine is located on a separate domain. The diagram below shows the network.

¹² <http://jakarta.apache.org/tomcat>

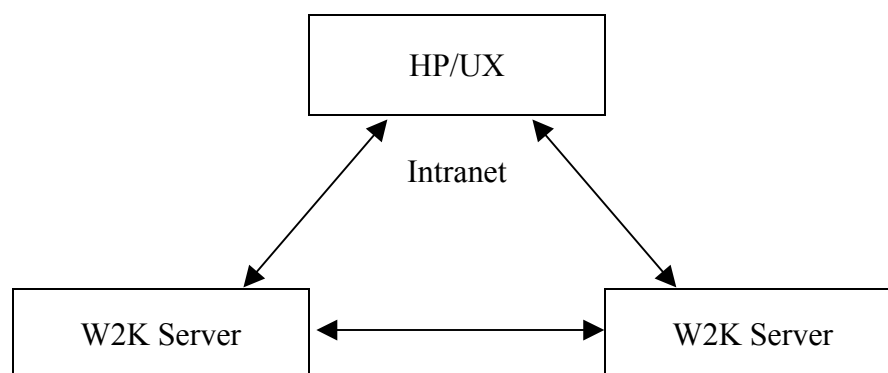


Diagram: Test machines network

Machine Specifications

Each of the W2K machines is a 300 MHz Pentium 2 workstation with ample memory on the workstation performing the EFM role so that no page swapping occurs during the testing (310 MB). The highest memory load observed during the testing was less than 200MB. The HP/UX machine is a server class machine with dual L2000 PA RISC Processors, and ample memory so that no page swapping occurs during the testing.

Performance Testing the Prototype

Goals

Our stress testing including the following goals:

- a.) Perform multiple concurrent filings with one payload file per filing over both the Internet and the intranet, using HP/UX to windows and vice versa. Perform the same test with 50 payload files per filing.
- b.) LA County District Attorneys' Office files 70,000 felonies and 400,000 misdemeanors per year. Therefore, we assessed our minimal performance benchmark at 5 filings/minute (1 filing/20 seconds), assuming 265 workdays, at 6 hours a day and the aforementioned volume. We assumed a 300% peak time to assess an upper target of 15 filings/minute (1 filing/4 seconds).
- c.) Assume all files will need to be virus checked and validated against our custom Court Filing schema

Procedure

For each test, we ran 25 sequential identical messages, with no pause between each send. We measured:

- The average time it took to send each message
- The average time it took to unwrap the ebXML messages

- The average time it took to validate and unwrap the CF messages
- The average overall time it took to send/receive/unwrap/validate all messages

For each batch of test sends, a different complaint filing was attached as an ebMS MIME attachment. Three batches were run consisting of:

- 1.) Our CF1.1 based filing envelope with a base 64 encoded copy of a PDF filing as its payload
- 2.) Our CF1.1 based filing envelope with base 64 encoded copies of a PDF filing and 5 documents comprising a JXDD incident report¹³ as its payload.
- 3.) Our CF1.1 based filing envelope with base 64 encoded copies of a PDF filing and 50 documents comprising 10 JXDD incident reports as its payload.

Between batches, files were removed from the inbox of the receiving machine, its database cleared, and Tomcat restarted.

The following tables enumerate our performance results.

¹³ The sample incident report, sponsored and developed by the Los Angeles County Sheriff's Dept is available here: http://it.ojp.gov/jsr/public/viewDetail.jsp?sub_id=189

Table 1: W2K to HP/UX across intranet- avg. ping: 11 ms					
Payload	Message Size ¹⁴	Send (W2K)	Unwrap (HP/UX)	Validate (HP/UX)	Overall ¹⁵
Felony Complaint PDF	115 KB	4.23 sec	.03 sec	2.87 sec	5.25 sec
Felony Complaint PDF plus Incident Report ¹⁶	297 KB	9.20 sec	.08 sec	2.84 sec	9.08 sec
Felony Complaint PDF plus 10 Incident Reports	1,943 KB	59.18 sec	.74 sec	3.60 sec	52.20 sec

Table 2: HP/UX to W2K across intranet- avg. ping: 13 ms					
Payload	Message Size	Send (HP/UX)	Unwrap (W2K)	Validate (W2K)	Overall
Felony Complaint PDF	115 KB	3.55 sec	.06 sec	7.07 sec	9.54 sec
Felony Complaint PDF plus Incident Report	297 KB	8.83 sec	.16 sec	7.08 sec	12.46 sec
Felony Complaint PDF plus 10 Incident Reports	1,943 KB	55.71 sec	1.73 sec	7.81 sec	48.77 sec

Table 3: 2 W2K machines to HP/UX across intranet- avg. ping: 10ms					
Payload	Payload Size	Send ¹⁷ (W2K)	Unwrap (HP/UX)	Validate (HP/UX)	Overall
Felony Complaint PDF	115 KB	4.19 sec	.05 sec	2.41 sec	4.91 sec
Felony Complaint PDF plus Incident Report	297 KB	9.9 sec	.08 sec	2.50 sec	8.85 sec
Felony Complaint PDF plus 10 Incident Reports	1,943 KB	60.73 sec	.60 sec	2.83 sec	41.06 sec

¹⁴ "Payload Size" is the disk space required to store a CF1.1 based envelope with each payload embedded within it as a separate base 64 encoded MIME attachment.

¹⁵ "Overall" is derived by taking a period of time from when the first file begins receipt to the time the last file is validated and dividing that period by the total number of files sent.

¹⁶ Incident Report includes 5 individual files including the .xml file along with two sets of supporting .xsl, and .xsd files

¹⁷ "Send" represents average time it took each W2K machine to send the complaint averaged together.

Virus scan performance

We placed 100 of our Felony Complaint XML documents into a single directory and ran McAfee's VirusScan on the directory. The contents of our test directory included 50 copies of our 115 KB filing with 1 PDF attachment, and 50 copies of our 297 KB filing with 1 PDF attachment and 5 XML attachments comprising an incident report. We configured VirusScan to scan all files as well as all compressed files. The results of this scan showed an average scan time of .07 seconds per file.

Results

Examining the data, two factors appear as the strongest performance limiters- the rate at which filings can be sent by the EFP, and the rate at which the EFM can validate them as file size increases. As expected, validation appears to be dependent on hardware capabilities, as the HP/UX server was able to validate faster than the W2K workstations. We monitored memory and CPU consumption during the testing process, and memory remained well under the machines available RAM at any given time while the CPU was generally pegged at 100% on the W2K workstations.

Overall, we neared our upper target benchmark of processing 15 filings/minute with a rate of 12 filings/minute when multiple senders filed to the HP/UX server (Table 3), the most "real-world" resembling configuration. The EFM processes the 115KB filing nearly twice as fast when two machines are simultaneously filing, so it is not clear that the EFM's processor is fully taxed nor that it could not process faster. Even using the slowest configuration (Table 2), we achieved our minimum benchmark of 5 filings/minute.

Final Notes

Summary

Our goal was to build and examine a prototype system for the electronic filing of criminal complaint documents using emerging technologies. To that end, we found that with some modification, a system can be created that is capable of processing an approximately real-world throughput of e-filings using ebXML and a filing schema based on the Court Filing 1.1 specification and the JXDD 3.0.

While this prototype does show feasibility, it also details a number of points that we would recommend be addressed in the CF1.1 and JXDD specifications.

Given the performance and stability of ebXML messaging, it seems that ebMS can be a good foundation for any future complaint filing system as the specifications for describing and encapsulating the filing information itself mature.

Credits

ISAB

Founded in 1982, the Los Angeles County Information Systems Advisory Body (ISAB) is a chartered organization dedicated to the development of Criminal Justice Enterprise technology standards and solutions in collaboration with the member agencies. The parent organization, Countywide Criminal Justice Coordinating Committee (CCJCC), provides executive sponsorship for ISAB technology initiatives. ISAB membership includes the Sheriff, District Attorney, Chief Probation Officer, City Attorney, Coroner, Board of Supervisors, Public Defender, Alternate Public Defender, Clerk of the Superior Court and related criminal justice partners from the State and Federal level. Initiatives include videoconferencing, justice integration middleware acquisition and support, justice enterprise Electronic Document Management System suite (EDMS) and initiation of numerous multi-agency integration projects.

LA County District Attorney Office

The District Attorney Office files approximately 70,000 felonies and 400,000 misdemeanors per year. This caseload includes providing numerous contract cities in LA County with Prosecution services. The District Attorney Office has long been a champion of technology initiatives to promote inter-agency exchange of criminal data. The DA office fully supports e-filing initiatives and has provided technical and business analysis support staff for development of the Justice XML criminal complaint.

Sierra Systems

Sierra Systems is a leading professional services firm specializing in providing high-quality, cost-effective business information systems solutions. Since its establishment in 1966, the Company has been delivering the benefits of information technology to clients in both the private and public sectors. Sierra Systems has over 900 employees in locations throughout the United States and Canada:

- Calgary
- Dallas
- Edmonton
- Halifax
- Hartford
- Los Angeles
- Olympia
- Ottawa
- San Diego
- Seattle
- Toronto
- Vancouver
- Victoria
- Washington, DC
- Winnipeg

Sierra Systems' mission is "to enhance the competitive position of our clients through the implementation of information technology-based business solutions."

Appendix A - Resources

GTRi – <http://justicexml.gtri.gatech.edu>

OASIS - <http://www.oasis-open.org/home/index.php>

EBXML - <http://www.ebxml.org/>

FreeBXML - <http://www.freebxml.org/>

LegalXML - <http://www.legalxml.org/>

Appendix B - Sample XML Filing Response

```
<?xml version = "1.0" encoding = "UTF-8"?>
```

```
<!--
```

This is a sample response to the petition filing in Appendix C. It is an ACKNOWLEDGEMENT of receipt accompanied with REJECTION of one of the received documents. It was generated by hand, not by an Electronic Filing Manager.

```
-->
```

```
<legalEnvelope xmlns="http://www.legalXML.org/"
  xmlns:jdd="http://www.it.ojp.gov/jxdd/prerelease/3.0.0.1"
  xmlns:usps="http://www.it.ojp.gov/jxdd/prerelease/usps_states/1.0.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.legalXML.org/ courtfiling11 2002 07 22.xsd
  http://www.it.ojp.gov/jxdd/prerelease/3.0.0.1 jxdds 3.0.0.1.xsd
  http://www.it.ojp.gov/jxdd/prerelease/usps_states/1.0.0.0
  usps_states_1.0.0.0.xsd">

  <messageIdentification>TA001071-23288103</messageIdentification>
  <creation>
    <dateTime>
      <date>2003-09-09</date>
      <time>22:21:00Z</time>
    </dateTime>
  </creation>

  <legal>
    <courtFiling>
      <confirmation>
        <timeStamp>
          <dateTime>
            <date>2003-09-09</date>
            <time>22:21:00Z</time>
          </dateTime>
        </timeStamp>
        <confirmationInformation filingDisposition="acknowledged"
          refersTo="filing.1">
          <leadDocumentDisposition filingDisposition="rejected"
            refersTo="TA001071-23288103.pdf">
            <timeStamp>
              <dateTime>
                <date>2003-09-09</date>
                <time>22:21:00Z</time>
              </dateTime>
            </timeStamp>
            <courtDocumentReference>
              http://efm.domain.com/documentName
            </courtDocumentReference>
          </leadDocumentDisposition>
        </confirmationInformation>
      </confirmation>
    </courtFiling>
  </legal>
</legalEnvelope>
```

Appendix C - Sample XML Filing (Felony Complaint w/o Attachments)

```
<?xml version = "1.0" encoding = "UTF-8"?>

<!--
Felony Complaint using hybrid CF1.1/JXDD schema
-->

<legalEnvelope xmlns="http://www.legalXML.org/"
  xmlns:jdd="http://www.it.ojp.gov/jxdd/prerelease/3.0.0.1"
  xmlns:usps="http://www.it.ojp.gov/jxdd/prerelease/usps_states/1.0.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.legalXML.org/ courtfiling11 2002 07 22.xsd
  http://www.it.ojp.gov/jxdd/prerelease/3.0.0.1 jxdds_3.0.0.1.xsd
  http://www.it.ojp.gov/jxdd/prerelease/usps_states/1.0.0.0
  usps_states_1.0.0.0.xsd">

  <messageIdentification>TA001071-23288103</messageIdentification>
  <creation>
    <dateTime>
      <!-- date this xml document was created -->
      <date>2003-09-09</date>
      <time>00:00:00Z</time>
    </dateTime>
  </creation>

  <legal>
    <courtFiling>
      <filing>
        <jdd:CaseParticipants>
          <!-- defendants, attorneys, etc. -->
          <jdd:CaseDefendantActor jdd:id="defendant.1">
            <jdd:Person>
              <jdd:PersonName>
                <jdd:PersonGivenName>
                  Jose Diego
                </jdd:PersonGivenName>
                <jdd:PersonSurName>
                  Garcia
                </jdd:PersonSurName>
              </jdd:PersonName>
              <jdd:PersonBirthDate>
                1969-11-13
              </jdd:PersonBirthDate>
              <!-- CII No. -->
              <jdd:PersonAssignedIDDDetails>
                <jdd:PersonStateID>
                  008565663
                </jdd:PersonStateID>
                <jdd:PersonOtherID jdd:IDTypeText="Defendant number">
                  01
                </jdd:PersonOtherID>
              </jdd:PersonAssignedIDDDetails>
            </jdd:Person>
          </jdd:CaseDefendantActor>

          <jdd:CaseInitiatingActor jdd:id="plaintiff.1">
            <jdd:Organization>
              <jdd:OrganizationName>
                The People of the State of California
              </jdd:OrganizationName>
            </jdd:Organization>
          </jdd:CaseInitiatingActor>
        </jdd:CaseParticipants>
      </filing>
    </courtFiling>
  </legal>
</legalEnvelope>
```

```

<jdd:CaseOfficial jdd:id="da.1">
  <jdd:PersonName>
    <jdd:PersonGivenName>
      Steve
    </jdd:PersonGivenName>
    <jdd:PersonSurName>
      Cooley
    </jdd:PersonSurName>
  </jdd:PersonName>
  <jdd:CaseOfficialRoleText>
    District Attorney
  </jdd:CaseOfficialRoleText>
</jdd:CaseOfficial>

<jdd:CaseOfficial jdd:id="deputy.1">
  <jdd:PersonName>
    <jdd:PersonGivenName>
      Debbie
    </jdd:PersonGivenName>
    <jdd:PersonSurName>
      Barton
    </jdd:PersonSurName>
  </jdd:PersonName>
  <jdd:CaseOfficialRoleText>
    Deputy District Attorney
  </jdd:CaseOfficialRoleText>
  <jdd:CaseOfficialRoleText>
    Charge Filer
  </jdd:CaseOfficialRoleText>
</jdd:CaseOfficial>

</jdd:CaseParticipants>

<filingInformation id="filing.1">
  <jdd:Case>
    <jdd:CaseTrackingID>
      TA001071
    </jdd:CaseTrackingID>
    <!-- da case number -->
    <jdd:CaseOtherID jdd:IDTypeText="DA case number">
      23288103
    </jdd:CaseOtherID>

    <jdd:CaseCourt jdd:id="court.1">
      <jdd:OrganizationAddress>
        <jdd:AddressCityName>
          Compton
        </jdd:AddressCityName>
      </jdd:OrganizationAddress>
      <jdd:CourtName>
        Superior Court of the State of California
      </jdd:CourtName>
    </jdd:CaseCourt>

    <!-- case events -->
    <jdd:CaseCourtEvent>
      <!-- Custody R'tn Date -->
      <jdd:CourtEventAppearance>
        <jdd:CourtAppearanceDate>
          2003-10-01
        </jdd:CourtAppearanceDate>
      </jdd:CourtEventAppearance>
    </jdd:CaseCourtEvent>

```

```

<jdd:CaseParticipants>
  <jdd:CaseDefendantActor
    jdd:id="defendant.1.2"
    jdd:ref="defendant.1" />
  <jdd:CaseInitiatingActor
    jdd:id="plaintiff.1.1"
    jdd:ref="plaintiff.1" />
  <jdd:CaseOfficial
    jdd:id="da.1.2"
    jdd:ref="da.1" />
  <jdd:CaseOfficial
    jdd:id="deputy.1.1"
    jdd:ref="deputy.1" />
</jdd:CaseParticipants>
</jdd:Case>
</filingInformation>
<leadDocument id="TA001071-23288103">
  <documentInformation>
    <jdd:CaseParticipants>
      <!-- everyone involved, again -->
      <jdd:CaseDefendantActor
        jdd:id="defendant.1.3"
        jdd:ref="defendant.1" />
      <jdd:CaseInitiatingActor
        jdd:id="plaintiff.1.2"
        jdd:ref="plaintiff.1" />
      <jdd:CaseOfficial
        jdd:id="da.1.3"
        jdd:ref="da.1" />
      <jdd:CaseOfficial
        jdd:id="deputy.1.2"
        jdd:ref="deputy.1" />
    </jdd:CaseParticipants>
    <jdd:Submission>
      <!-- when the attached document was submitted -->
      <jdd:SubmissionSubmittedDate>
        2003-09-09
      </jdd:SubmissionSubmittedDate>
    </jdd:Submission>
    <jdd:Document>
      <jdd:DocumentDescriptiveMetadata>
        <jdd:DocumentTitleText>
          Felony Complaint
        </jdd:DocumentTitleText>
        <!-- Operator -->
        <jdd:DocumentCreator>
          <jdd:PersonTypeElement>
            <jdd:PersonName>
              <jdd:PersonNameInitialsText>
                GI
              </jdd:PersonNameInitialsText>
            </jdd:PersonName>
          </jdd:PersonTypeElement>
        </jdd:DocumentCreator>
      </jdd:DocumentDescriptiveMetadata>
    </jdd:Document>
  </documentInformation>
</leadDocument>

```

```

<criminal>
  <jdd:Actor
    jdd:id="defendant1.1"
    jdd:ref="defendant.1" />
  <jdd:Charge>
    <jdd:ChargeSequenceID>
      1
    </jdd:ChargeSequenceID>

    <jdd:ChargeDescriptionText>
      Willful, Deliberate, Premeditated Murder
    </jdd:ChargeDescriptionText>

    <jdd:ChargeClassification>
      <jdd:ChargeSentenceRangeText>
        Life
      </jdd:ChargeSentenceRangeText>
    </jdd:ChargeClassification>

    <jdd:ChargeStatute>
      <jdd:StatuteCodeID>
        PC
      </jdd:StatuteCodeID>
      <jdd:StatuteCodeSectionID>
        187(a)
      </jdd:StatuteCodeSectionID>
      <jdd:StatuteLevelText>
        F
      </jdd:StatuteLevelText>
    </jdd:ChargeStatute>

    <jdd:ChargeSubject
      jdd:id="defendant1.4"
      jdd:ref="defendant.1" />
  </jdd:Charge>

  <jdd:Arrest>
    <jdd:ArrestOfficial>
      <jdd:PersonName>
        <jdd:PersonSurName>
          Hernandez
        </jdd:PersonSurName>
        <jdd:PersonNameInitialsText>
          J
        </jdd:PersonNameInitialsText>
      </jdd:PersonName>
      <jdd:EnforcementOfficialBadgeID>
        983234
      </jdd:EnforcementOfficialBadgeID>
    </jdd:ArrestOfficial>
  </jdd:Arrest>
</criminal>

```

```

        <jdd:ArrestAgency>
            <jdd:OrganizationName>
                Cal State Dominguez Hills
            </jdd:OrganizationName>
        </jdd:ArrestAgency>

        <jdd:ArrestBailRecommendationText>
            $500,000
        </jdd:ArrestBailRecommendationText>

    </jdd:Arrest>

    <jdd:Booking>
        <jdd:BookingAgencyRecordID>
            004816010
        </jdd:BookingAgencyRecordID>
    </jdd:Booking>

    <jdd:Offense>
        <jdd:ActivityDate>
            2003-07-07
        </jdd:ActivityDate>

        <jdd:IncidentVictim>
            <jdd:PersonName>
                <jdd:PersonGivenName>
                    Devon
                </jdd:PersonGivenName>
                <jdd:PersonSurName>
                    White
                </jdd:PersonSurName>
            </jdd:PersonName>
        </jdd:IncidentVictim>

        <jdd:IncidentLocation>
            <jdd:LocationName>
                County of Los Angeles
            </jdd:LocationName>
        </jdd:IncidentLocation>

    </jdd:Offense>
</criminal>

</documentInformation>
<documentContent id="TA001071-23288103.pdf" mimeType="text/xml">
    ... base64 encoded attachment normally goes in here ...
</documentContent>
</leadDocument>
</filing>
</courtFiling>
</legal>
</legalEnvelope>

```


Appendix D - LA County Sample Felony Complaint (BLOB)

SUPERIOR COURT OF THE STATE OF CALIFORNIA
FOR THE COUNTY OF LOS ANGELES

THE PEOPLE OF THE STATE OF CALIFORNIA, v. 01 JOSE DIEGO GARCIA (11/13/1969)	Plaintiff, Defendant(s).	CASE NO. TA001071 FELONY COMPLAINT
----------------------------------------------------------------------------------------------	---------------------------------	--------------------------------------------------

The undersigned is informed and believes that:

COUNT 1

On or about July 7, 2003, in the County of Los Angeles, the crime of WILLFUL, DELIBERATE, PREMEDITATED MURDER, in violation of PENAL CODE SECTION 187(a), a Felony, was committed by JOSE DIEGO GARCIA, who did unlawfully and with malice aforethought attempt to murder DEVON WHITE, a human being.

It is further alleged that the aforesaid attempted murder was committed willfully, deliberately and with premeditation within the meaning of Penal Code section 664(a) and is a serious felony pursuant to Penal Code section 1192.7(c).

"NOTICE: Conviction of this offense will require you to provide specimens and samples pursuant to Penal Code section 296. Willful refusal to provide the specimens and samples is a crime."

It is further alleged that in the commission and attempted commission of the above offense, the said defendant(s), JOSE DIEGO GARCIA, personally used a firearm(s), to wit: rifle, within the meaning of Penal Code sections 1203.06(a)(1) and 12022.5(a)(1) also causing the above offense to become a serious felony pursuant to Penal Code section 1192.7(c)(8) and a violent felony within the meaning of Penal Code section 667.5(c)(8).

* * * * *

**SUPERIOR COURT OF THE STATE OF CALIFORNIA
FOR THE COUNTY OF LOS ANGELES**

THE PEOPLE OF THE STATE OF CALIFORNIA, Plaintiff, v. 01 JOSE DIEGO GARCIA (11/13/1969) Defendant(s).

CASE NO. TA001071

FELONY COMPLAINT

The undersigned is informed and believes that:

COUNT 1

On or about July 7, 2003, in the County of Los Angeles, the crime of WILLFUL, DELIBERATE, PREMEDITATED MURDER, in violation of PENAL CODE SECTION 187(a), a Felony, was committed by JOSE DIEGO GARCIA, who did unlawfully and with malice aforethought attempt to murder DEVON WHITE, a human being.

It is further alleged that the aforesaid attempted murder was committed willfully, deliberately and with premeditation within the meaning of Penal Code section 664(a) and is a serious felony pursuant to Penal Code section 1192.7(c).

"NOTICE: Conviction of this offense will require you to provide specimens and samples pursuant to Penal Code section 296. Willful refusal to provide the specimens and samples is a crime."

It is further alleged that in the commission and attempted commission of the above offense, the said defendant(s), JOSE DIEGO GARCIA, personally used a firearm(s), to wit: rifle, within the meaning of Penal Code sections 1203.06(a)(1) and 12022.5(a)(1) also causing the above offense to become a serious felony pursuant to Penal Code section 1192.7(c)(8) and a violent felony within the meaning of Penal Code section 667.5(c)(8).

* * * * *

FELONY COMPLAINT -- ORDER HOLDING TO ANSWER -- P.C. SECTION 872

It appearing to me from the evidence presented that the following offense(s) has/have been committed and that there is sufficient cause to believe that the following defendant(s) guilty thereof, to wit:

(Strike out or add as applicable)

<u>JOSE DIEGO GARCIA</u>				
<u>Count</u>		<u>Charge</u>	<u>Special</u>	<u>Alleg.</u>
<u>No.</u>	<u>Charge</u>	<u>Range</u>	<u>Allegation</u>	<u>Effect</u>
1	PC 187(a)	Life	PC 12022.5(a)(1)	+3-4-10

I order that the defendant(s) be held to answer therefor and be admitted to bail in the sum of:

JOSE DIEGO GARCIA _____ Dollars

and be committed to the custody of the Sheriff of Los Angeles County until such bail is given. Date of arraignment in Superior Court will be:

JOSE DIEGO GARCIA _____ in Dept _____

at: _____ A.M.

Date: _____

Committing Magistrate