7 Steps to Electronic Filing with Electronic Court Filing 4.0

The 7 Steps found within this Quick Start Guide will assist you with the minimum requirements to implement e-Filing with the OASIS Electronic Court Filing 4.0 Specification.

QUICK START GUIDE

THE ECF SPECIFICATION ALLOWS YOU TO:

- Standardize integration methods in your e-Filing implementation with XML
- Integrate with any potential e-Filing Service Provider or share e-Filing data between systems or with partners
- Setup a single method of processing data related to e-Filing
- Find out how to implement legal service in your e-Filing application

PREPARED BY:

OASIS LegalXML Court Filing Technical Committee
In conjunction with the release of the Electronic Court Filing 4.0 Specification

INTRODUCTION

In the software industry today, and especially in the global justice space, there continues to be a strong movement toward the standardization of data definition and exchange methods, and it is often the desire of administrators and technologists throughout federal, state, local, and tribal governments to apply those standards. In reality, however, it is often overwhelming to review and fully comprehend the documentation accompanying those standards. This document attempts to minimize that factor for the OASIS Electronic Court Filing (ECF) specification, and allows the reader to completely understand what is required to implement the specification in their environment.

While the ECF specification covers a wide range of use cases and possible data exchange transactions, only a small sub-set of those transactions are required in order to implement the specification. In fact, in 7 easy steps we can break down the tasks necessary for a fully compliant ECF implementation.

The steps for implementing ECF are:

1. Identify e-Filing Service Provider(s)
2. Identify an e-Filing Manager
3. Choose to implement ECF 4.0
4. Develop your Court Policy
5. Understand MDE's, Operations, and Messages
6. Choose a Service Interaction Profile
7. Develop and Implement

That's it!? But wait a minute, the first two steps have nothing to do with ECF, do they? In fact, they do. It turns out implementing ECF does have at least two prerequisites for an end-to-end implementation. There must be 1) a system that produces the filing, and 2) a system that receives the filing. You will learn more as you read through the 7 easy steps.

STEP 1. IDENTIFY E-FILING SERVICE PROVIDER(S)

For those who are unfamiliar with the ECF specification, there is common misunderstanding that the specification itself is a complete e-Filing solution. This is not the case. Rather, ECF is the solution that allows those systems or entities participating in the e-Filing process to communicate and exchange data with one another. The primary system utilized to prepare and submit court filings electronically is known as an Electronic Filing Service Provider or EFSP. Your first step to implementing the ECF specification will be to identify at least one EFSP. That's correct, at LEAST one EFSP is required, but the ECF specification allows for multiple EFSPs if desired.

Electronic Filing Service Providers are available to e-Filing implementers in various forms and fashions, including but not limited to the following:

- E-Filing Vendor – several commercial e-Filing systems are available in the market place, of which one or more could be selected for use in your e-Filing implementation.
- In House – many courts have developed their own e-Filing systems that include an EFSP.
- System Customization – as an example, document generation software could be customized to allow the user to automatically generate and submit documents for filing.

Alright, so perhaps Step 1 is not necessarily "easy", but if you are thinking about implementing e-Filing or the ECF specification, odds are that you have either already identified an EFSP or have given thought to one of the above options.

One of the enormous benefits of utilizing the ECF specification is that it does not restrict you to a specific EFSP. Any or all of these EFSPs could be implemented. In fact, utilizing the ECF specification leaves the implementer's environment open for additional EFSP integrations, and it could easily be argued that this is necessary in most courts. Even in a single vendor e-Filing implementation, the court will likely have a need to directly integrate with other government agencies (i.e. Prosecutor's offices, public defender offices, law enforcement offices, lower courts, appellate courts, among others) allowing them to electronically file documents with the court directly through the automated systems of that agency.

As you will learn in Step 4, in ECF specification terms the EFSP will be responsible for acting as the Filing Assembly Major Design Element (MDE) and generating the XML core filing message, for submission to the court as an electronic filing.

STEP 2. IDENTIFY AN E-FILING MANAGER

Now that you have an EFSP, the next step is to determine the manner by which the court would like to consume and manage electronic filings generated by that EFSP. Applications implemented for this purpose are commonly referred to as the Electronic Filing Manager or EFM, and often vary widely in the level of functionality they provide.

As with EFSPs, Electronic Filing Managers are also available to e-Filing implementers in various forms and fashions. In addition, your selection of an EFM may be highly dependent on what EFSP(s) have been identified. Commercial vendors typically offer what would be considered EFM functions within their software and may or may not require its use should you select that vendor's system. Or, the court's case management system software may have built-in EFM capabilities. The court also has the option to custom develop their own EFM software. The beauty of the ECF specification is that it doesn't matter which EFM option you choose, it is designed to work with any of them.

With reference to ECF specification terminology, the EFM will act as the Filing Review MDE and will be responsible for interacting with the Filing Assembly MDE provided by the EFSP to communicate the acknowledgement and acceptance of new filings, and to provide updates on the status of said filings. Specific XML messages have been defined for this purpose.

It is important to also acknowledge that there is an expectation that additional applications will exist in a court's e-Filing enviroment to complete an end-to-end e-Filing system. These applications include a Case Management System (CMS) and a Document Management System (DMS). As with EFSPs and EFMs these applications may exist in a variety of formats, but must be considered and available to completely process an e-Filing utilizing the ECF specification.

STEP 3. CHOOSE TO IMPLEMENT ECF 4.0

Now that an EFSP and EFM have been identified, there is a high likelihood that you will need to choose a method by which to either integrate those applications with one another, or with others applications within your e-Filing system. In many cases, multiple components of an e-Filing system exist within the same application and therefore do not require integration with each other; however it would be extremely rare for ALL components of an end-to-end e-Filing system to exist completely within the same application. As a result, there will be a need for the components of these separate applications to communicate with one another to fully process an electronic filing.

To further illustrate this point, consider the following examples of possible e-Filing systems:

- A vendor provides both your EFSP and EFM, but the Court hosts its own Document Management (DMS) and Case Management Systems (CMS).
- A vendor provides an EFSP, but the Court hosts its own EFM within its CMS.
- A Court developed its own EFSP and EFM separate of the CMS and DMS.

These examples demonstrate the variety of options that are available when constructing an e-Filing system, and that a complete e-Filing solution will almost always require some integration points. This integration may be the EFSP and EFM exchanging data with one another, or the EFM communicating with external DMS and CMS applications. If we are assuming this as fact, then we can also assume that a method for integration between these applications is vital to the success of a complete e-Filing implementation.

This is exactly why the ECF 4.0 specification has come into existence. The authors of the specification saw this inevitable truth, and the need to develop an integration method that is standardized and repeatable, allowing courts and vendors alike

to develop applications capable of interacting with multiple external application in a singular way for the purpose of e-Filing.

STEP 4. DEVELOP YOUR COURT POLICY

Perhaps most important step in implementing the ECF specification is to define the ECF Court Policy. The court policy is integral to the e-Filing process as it defines a myriad of requirements the Filing Assembly MDE must consider. In defining a court policy, the Court must give considerable thought to exactly how comprehensive an e-Filing application it wishes to implement. Will it include a single or multiple case type, allow for electronic service, or provide access to other case documents? Answers to these questions should be defined in the court policy.

Additionally, the Court will need to determine how e-Filed documents will be "signed" and define a document signature profile. The ECF specification allows for digital signatures and XML signatures, among others, or in the absence of more formal signature technologies a "null" signature profile. Most courts have found it perfectly acceptable to implement the null signature profile along with a "/s/ Filer Name" on the document signature line, as, in combination with the e-Filing process itself, it demonstrates the intent to sign and file.

Further demonstrating the importance of court policy, it will define codes associated with specific courts, or code lists an EFSP might be required to use to associate data it is submitting within the core filing message. Courts are required to provide their court policy in two formats; 1) Human-Readable, and 2) Machine-Readable.

Human-Readable Court Policy is a textual document publishing the court's rules and requirements for electronic filing. To be compliant with the ECF 4.0 specification, each court MUST publish a human-readable court policy that MUST include each of the following:

- The unique court identifier.
- The location of the machine-readable court policy.
- A definition of what constitutes a "lead document" in the court.
- A description of how the <filingPartyID> and <filingAttorneyID> are to be maintained during electronic communications regarding the case.
- A description of how the court processes (dockets) matters.
- A description of any instances in which the court will mandate an element that the ECF 4.0 schema makes optional.
- A description of any restrictions to data property values other than code list restrictions.
- Any other rules required for electronic filing in the court.

Machine-Readable Court Policy is an ECF 4.0 message that describes the features of the ECF 4.0 implementation supported by the implementing court, the court's code lists, and any other information a Filing Assembly MDE would need to know in order to electronically file successfully into that court. Machine-readable court Policy includes structures for identifying run-time and development-time policy information.

Run-time information includes information that will be updated from time to time, such as code lists (e.g., acceptable document types, codes for various criminal charges, and civil causes of action) and the court's public key for digital signatures and encryption.

Development-time information includes court rules governing electronic filing that are needed at the time an application is developed but which are not likely to change. These include:

- The service interaction profile(s) that the court supports (see step 6).
- The MDEs, query operations and case types supported by the court's ECF 4.0 system.
- Whether a court will accept the filing of a URL in lieu of the electronic document itself.
- Whether the court accepts documents requiring payment of a filing fee.
- Whether the court accepts electronic filing of sealed documents.
- Whether the court accepts multiple (batch) filings.
- The court-specific extensions to the ECF 4.0 specification, including the required elements.
- The maximum sizes allowed for a single attachment and a complete message stream.

Thankfully, implementing courts don't have to stress over how to format their machine readable court policy, as the ECF specification provides the xml structure to be used. Here's a short snippet as an example:

```
<j:CaseCourt>
<nc:OrganizationIdentification>
<nc:IdentificationID>ESS-SC</nc:IdentificationID>
</nc:OrganizationIdentification>
 <j:CourtName>Essex Superior Court</j:CourtName>
 </j:CaseCourt>
```

Self explanatory, right? To those familiar with XML it should be, and to those who have never seen XML it should seem clear that the court's ID is "ESS-SC" and the court's name is "Essex Superior Court". The rest of the XML defined in the ECF Court Policy Response Message is similar.

Court Policy Tip

Write your Human-Readable court policy and completely define all the requirements with all the elements described above within that policy, then work with your technical team members to translate that policy into the Machine-Readable XML version.

STEP 5. UNDERSTAND MDE'S, OPERATIONS, AND MESSAGES

Perhaps the hardest step in implementing the ECF specification is understanding the process flow and its components, then correctly mapping the data the court finds necessary to process a filing with the XML schemas provided by the ECF specification. Gaining this understanding of minimum requirements for ECF should help tremendously. There are literally only three MDE's (in green), seven message interactions (in purple), and five operations (in blue) required to implement a fully compliant ECF e-Filing solution, and are briefly described here.

1. The Filing Assembly MDE will generate the Court Policy Query Message, Core Filing Message initiate the Get Policy and Review Filing operation.
2. The Filing Review MDE will generate, the Court Policy Response Message, the Record Docketing Message, the Review Filing Callback Message, and initiate the Record Filing and Notify Filing Review Complete operations.
3. The Court Record MDE will generate the Record Docketing Callback Message and initiate the Notify Docketing Complete operation.
4. For each message initiated, the receiving MDE shall generate the Message Receipt Message synchronously.

This means that in terms of mapping data to the ECF message schemas, the majority of the work will occur in these seven messages defined in the purple text. The specification provides detailed schemas for each message, so that task is simply determining where to insert the data your implementation requires within the schema. Let's look at an example and code snippet from each message.

The CourtPolicyQueryMessage is a simple message generated by the Filing Assembly MDE to get the requirements of the court it is attempting to file in. The key to this transaction for the court is knowing which Filing Assembly MDE is making the request, therefore the query message defines the requestor. Here's a snippet:

```
<ecf:QuerySubmitter>
<ecf:EntityPerson> <nc:PersonOtherIdentification>
<nc:IdentificationID>1</nc:IdentificationID>
<nc:IdentificationCategoryText>Vendor A</nc:IdentificationCategoryText>
</nc:PersonOtherIdentification>
</ecf:EntityPerson>
</ecf:QuerySubmitter>
```

In this example, the court assigned ID, "1", is defined as well as a textual description, "Vendor A", so that the court knows who to respond to with the CourtPolicyResponseMessage. In terms of mapping this indicates the court will need to assign an ID for each EFSP, and map that ID to these data elements in preparing for implementation. Here's a snippet of the court's response:

```
<DevelopmentPolicyParameters>
<SupportedCaseType>Civil</SupportedCaseType>
<FilingFeesMayBeApplicableIndicator>true</FilingFeesMayBeApplicableIndicator>
<EffectiveDate>
<nc:Date>2008-08-01</nc:Date>
</EffectiveDate>
</DevelopmentPolicyParameters>
```

This court policy response indicates that currently the court is only accepting filings in the Civil case type, where filing fees may be applicable, and effective 8/1/2008. This demonstrates clearly

how the court will map the case types for which it will allow e-Filing to occur within the court Policy message.

In this case, the Filing Assembly MDE now has the knowledge that it may generate a CoreFilingMessage for civil filings. In that message, it will define the document(s) to be filed. Here's a snippet of the message:

```
<FilingLeadDocument s:id="MotionToContinue.doc">
<nc:DocumentApplicationName>Microsoft Word
</nc:DocumentApplicationName>
<nc:DocumentDescriptionText>Motion to Continue
</nc:DocumentDescriptionText>
<nc:DocumentLanguageCode>eng
</nc:DocumentLanguageCode>
</FilingLeadDocument>
```

In this small snippet, we see that the document defined in this filing is a "Motion to Continue", it is the lead document, and it can be identified by the file name of "MotionToContinue.doc".

The Filing Review MDE should immediately respond to the submission of the CoreFilingMessage with a MessageReceiptMessage. The majority of this message repeats information provided by the submitter, but the important piece will indicate whether or not the message was received. Here's a snippet:

```
<message:Error>
  <message:ErrorCode>0</message:ErrorCode>
  <message:ErrorText>No error</message:ErrorText>
 </message:Error>
```

In the sample above, a code of "0" was returned indicating no errors occurred and the filing was received by the Filing Review MDE. The court will need to define a set of error codes for use in this message, and descriptions for each code, then map them to this message.

Once a determination has been made, either by automation or through manual intervention and review that the filing is good and will be accepted, the Filing Review MDE will initiate the Record Filing operation with the RecordDocketingMessage. This message should define the filing in a way that allows the Court Record MDE to fully process it and respond. Here's a snippet of that definition:

```
<ecf:DocumentMetadata>
<j:RegisterActionDescriptionText>MOT</j:RegisterActionDescriptionText>
<ecf:FilingAttorneyID>
<nc:IdentificationID>562455</nc:IdentificationID>
</ecf:FilingAttorneyID></ecf:DocumentMetadata>
```

The snippet here show's that the document has now been defined for specific data elements that will be recorded by the Court Record MDE, likely in the CMS and DMS. In this case "MOT" represents a code within the system for Motion, and the "562455" represents the state bar number of the attorney filing the document.

Once the Court Record MDE completes processing, it will initiate the Notify Docketing Complete operation and send RecordDocketingCallbackMessage to the Filing Review MDE. This message will contain the following snippet:

```
<nc:DocumentPostDate>
<nc:DateTime>2007-06-06T14:20:46.0Z</nc:DateTime>
</nc:DocumentPostDate>
...
<ecf:FilingStatus>
<nc:StatusDescriptionText>Without modification</nc:StatusDescriptionText>
<ecf:FilingStatusCode>Docketed</ecf:FilingStatusCode>
</ecf:FilingStatus>
```

In this instance, the Filing Review MDE now becomes aware that the document has been fully "Docketed", "Without Modification", at "14:20:46" on "6/6/2007". A mapping effort will need to occur so that data relevant to the court's CMS is positioned correctly within this schema.

This information can now be relayed full circle by the Filing Review MDE to the Filing Assembly MDE with the NotifyFilingReviewComplete operation ReviewFilingCallbackMessage. This message will confirm the status of the filing for the Filing Assembly MDE and would contain the following snippet:

```
<nc:DocumentFileDate>
<nc:DateTime>2007-06-06T14:20:46.0Z</nc:DateTime>
</nc:DocumentFileDate>...
<ecf:FilingStatus>
<nc:StatusDescriptionText>Without modification</nc:StatusDescriptionText>
<ecf:FilingStatusCode>Accepted</ecf:FilingStatusCode>
</ecf:FilingStatus>
```

With this information the Filing Assembly MDE can now update any data in the EFSP and notify the filer to indicate the document was "Accepted" by the court on "8/13/2007", "Without Modification".

This brief narration of the events, operations, and messages should demonstrate that while the ECF specification may appear complex, in reality the process is simple, and the messages are basic in nature. Mapping the data utilized in your systems to the ECF message schemas will be extremely important, especially when attempting to avoid confusion between systems on the meaning of an error code or status description.

STEP 6. CHOOSE A SERVICE INTERACTION PROFILE

With an understanding of the overall process flow and its components in place, the system is now in need of a method by with these components will communicate and exchange messages. The ECF specification defines this method as the Service Interaction Profile or SIP.

An ECF 4.0 SIP defines a transmission system that supports the functional requirements of electronic filing and the MDE operations and message structures, and implements certain non-functional requirements, but does not govern the content of messages. A service interaction profile will define how a message gets from the sending MDE to the receiving MDE in a given messaging framework.

The ECF technical committee has currently defined and accepted two possible SIPs that may be used in compliance with the ECF specification. This does not, however, preclude an implementer from defining and developing their own SIP for consideration and acceptance into the specification.

The Web Services Service Interaction Profile 2.0 specification defines a transmission system using the specifications described in the Web Services Interoperability (WS-I) Basic Profile 1.1, and WS-I Basic Security Profile 1.0 To utilize the web services profile, it simply requires that the appropriate web services be developed and made available for each MDE for the purpose of initiating each of the required operations, and submitting messages for consumption.

The Media Service Interaction Profile 1.0 specification defines a transmission system in which the sending MDE stores message transmissions on portable media (e.g., a compact disc) which is then physically transported to the receiving MDE where it is connected for retrieval of the message transmissions. This specification may be needed in the absence of an active network between the sending and receiving MDEs. While this is not the most popular SIP, it is a viable SIP that could accomplish the tasks involved in completely processing e-Filings. The recommendation for this SIP, however, would be that it only be used in emergency situations or perhaps in those situations where bulk filing may occur.

If neither of these approved SIPs makes sense for your environment, it is highly recommended that you define and develop your own SIP for implementation, and submit the SIP to the OASIS ECF technical committee for approval as a specification.

STEP 7. DEVELOP AND IMPLEMENT

Finally, all the pieces are in place to get to work. We've identified our EFSP(s) and EFM, we know we are going to use the ECF 4.0 specification, we understand the specification and the concept of MDE's and the specification's use of operations and messages, and we have selected a SIP that will allow the components to communicate with one another. This should now allow you to set a course to develop and implement your e-Filing solution. As always, the development and implementation process will be highly dependent upon the choices you have made, but in an overall sense, you will need to support the following operations.

GET POLICY - The Filing Assembly MDE MAY obtain a court's machine-readable court policy at any time by invoking the GetPolicy operation on the Filing Review MDE with the identifier for the court. The Filing Review MDE returns the machine-readable court policy in a synchronous response. This step may be omitted if the Filing Assembly MDE already has the current court policy.

REVIEW FILING - The Filing Assembly MDE MUST submit the filing to the court by invoking the ReviewFiling operation on the Filing Review MDE. The ReviewFiling operation includes messages for the core filing, for case type-specific information, for court-specific information, and for the filing payment. The Filing Review MDE responds synchronously with a receipt message that includes the filing identifier issued by the court.

RECORD FILING - If the clerk reviews and accepts the filing, the Filing Review MDE MUST invoke the RecordFiling operation on the Court Record MDE. The RecordFiling operation includes information from the ReviewFiling operation with any modifications or comments by the clerk. The Court Record MDE responds synchronously with an acknowledgement of the request.

NOTIFY DOCKETING COMPLETE - The Court Record MDE MUST invoke the NotifyDocketingComplete operation on the Filing Review MDE as a callback message to the RecordFiling operation to indicate whether the filing was accepted or rejected by the court record system. If the Court Record MDE rejected the filing, an explanation MUST be provided. If the Court Record MDE accepts the filing, the docketing information MUST be provided. The Filing Review MDE responds synchronously with an acknowledgement of the callback message.

NOTIFY FILING REVIEW COMPLETE - If the court rejects the filings or the Filing Review MDE receives the Notify Docketing Complete message, the Filing Review MDE MUST invoke the NotifyFilingReviewComplete on the Filing Assembly MDE as a callback message to the ReviewFiling operation to indicate whether the filing was accepted and docketed by the clerk and court record system. The operation MAY return the filed documents or links to the documents. If the filing included a payment and the filing was accepted by the court record system, a receipt for the payment MUST be included in the operation. The Filing Assembly MDE responds synchronously with an acknowledgement of the callback message.

LEGAL SERVICE MDE

Many would argue that an e-Filing implementation without the ability to electronically serve your pleadings as part of the electronic filing process is, in essence, an incomplete solution. There is validity to this argument, and it should be considered strongly as you plan for your e-Filing implementation. From the perspective of the ECF specification, it is important to note, that while implementing legal service is not a requirement of the specification it is sufficiently accounted for within the specification with the Legal Service MDE. In short, the Legal Service MDE allows an implementer to publish a base service list, which would be typically populated with party and attorney information from the court's case management system, for use by a Filing Assembly MDE or EFSP for the purpose of electronic service.

CONCLUSION

In the following section entitled "Important Links", we've provided the information to directly download the full ECF specification and correlated documents. As previously stated, the full specification can appear overwhelming, if referenced in light of the steps outlined here, you should find you can focus only on those sections required for an ECF compliant implementation, and therefore more clearly understand how to piece together your solution. In doing so, you should have an e-Filing environment that achieves the purpose of the specification in allowing you to integrate with multiple applications in a singular and standardized way.

IMPORTANT LINKS

Oasis ECF Technical Committee Web Site
http://www.oasis-open.org/committees/legalxml-courtfiling/

Full ECF 4.0 Specification
http://docs.oasis-open.org/legalxml-courtfiling/specs/ecf/v4.0/ecf-v4.0-spec/ecf-v4.0-spec-cd01.zip

ECF 4.0 Web Services Service Interaction Profile
http://docs.oasis-open.org/legalxml-courtfiling/specs/ecf/v4.0/ecf-v4.0-webservices-spec/ecf-v4.0-webservices-spec-cd01.zip