

**From:** Shane Durham, LexisNexis *File & Serve*  
**To:** LegalXML Electronic Court Filing Subcommittee  
**Regarding:** Proposed Changes to ECF 3.01 Schema  
**Date:** December 4, 2006

## **Introduction**

The purpose of this document is to propose revisions to the LegalXML ECF 3.01 schema. These proposed changes are based on lessons learned from an implementation of ECF during a project known as 'Bulk Filing for LexisNexis *File & Serve*'. That project's implementation of ECF has been previously described in a posting to the OASIS LegalXML listserv ( '*On the Use of ECF 3.0 by LexisNexis File and Serve 'Bulk Filing'*', posted 10/25/06 ). That posting was also accompanied by a modified ECF schema which can act as a suggestion for how most of the proposals in this document might be accomplished.

## **Enhance ECF 'policy' to support context restrictions**

LexisNexis' proposed changes for ECF policy have been previously described in a LegalXML posting. ( '*Issue and Proposed Resolution Regarding MDE Policy*', posted 11/10/06).

In a nutshell, it is proposed the schema be enhanced to allow developers to express restrictions for when certain codes may be included in a filing message, depending on what other values might be present in the same message, or broader contexts such as the functional permissions of the filer.

## **Enhance ECF to correct id/idref problem**

The current ECF schema has a bug where some data is to be expressed using an id/idref technique, such as party-to-attorney relationships, however the schema does not permit the programmer to express the 'id' part of that equation. It has already been noted by others that an 'id' data type must be added to those ECF nodes which express party or organization details.

## **Enhance ECF to express bar numbers as defined by GJXDM**

The ECF schema failed to include the formally defined GJXDM structures to express attorney bar numbers. These nodes should be included in ECF.

## **Enhance ECF to express multiple filing parties**

The ECF schema currently permits only one filing party to be expressed, per document. It should allow multiple filing parties for those instances where an attorney is working on behalf of multiple litigants.

## **Enhance ECF to permit the expression of filing-related fees.**

Note: Perhaps this is an oversight of LexisNexis, but we were unable to identify any ECF nodes to express the fees related to the processing of filing transactions.

The ECF schema should be enhanced so that a reviewed filing's fees may be expressed at both a filing/transaction level, and at a per-document level.

Similar changes are needed for the docketing messages which are unable to express the fees assigned by the court to an accepted filing. Likewise, the 'Fees Calculation' response message does not provide a sufficient structure to express filing fees. The schema describes a lump sum value, where it should support the description of a fee list, including multiple kinds (or types) of fees, as defined by the court's policy.

### **Enhance ECF to permit the expression of implementation-specific timestamps**

For LexisNexis *File & Serve*, it was relevant that an accepted filing (a.k.a. docket entry) express the filing's actual ReviewDate and Time, in addition to the official filing date and time legally recognized by the court.

Rather than request the ECF schema be enhanced to support this additional timestamp, LexisNexis proposes the schema be enhanced such that any number of implementation-specific timestamps might be expressed for a filing (and/or filing document). This is a flexible approach that can be handy in many ways.

### **Enhance ECF to not require court events to be expressed in the case details of an accepted filing.**

The ECF schema includes a node, 'DocketedCaseType', to express information about the case into which an accepted filing has been docketed. This node currently requires that one or more 'CourtEvents' be included in the case information. LexisNexis suggests the ECF schema be changed so that the case information can be expressed without having to include any information regarding the case's calendar.

### **Enhance ECF to express 'judgment amounts' related to collections cases**

For collections cases, during a new-case filing, a filer may express values representing the amount of money in dispute, and the legal categorizations of that money (examples: base amount, accumulated interest, legal fees, etc). In some courts this is known as the case's 'prayed for' values. Often these submitted values are referred to as 'default judgment values', since the court will frequently award them when disposing of those cases where the defendant does not participate in the case. (e.g. 'Since the defendant has not chosen to appear or to file an answer, a default judgment has been granted in favor of the plaintiff for the requested amounts of *yadda-yadda*.')

Once filed, these requested 'judgment' amounts are tracked as a property of the case, which may be modified later by additional filings (sometimes known as 'stipulations').

Later, once the case is decided, the court will assign 'final judgment amounts' to the case. These represent the awarded values (if any) to the plaintiff. With *File & Serve*, the 'final judgment amounts' are managed as a distinct set of values from those which were submitted at the start of the case. The user may view both the requested judgment amounts and the awarded judgment amounts.

LexisNexis suggests the ECF schema be enhanced to permit judgment amounts to be expressed in both new-case and existing-case filings. We also suggest the schema be enhanced so that a case may express both 'default' (requested) and 'final' (awarded) judgment amounts.

### **Enhance ECF to better express party-attorney relationships**

The ECF schema includes a node, 'CaseAttorneyRole', to express party-counselor relationships but its intended use is not very clear. It consists of 'n' number of IDREFs, without any indication of which 'ref' corresponds to a party and which corresponds to the counselor(s). Perhaps, this issue can be addressed with documentation, but, it would be better if the schema were modified such that the ambiguity was entirely eliminated. (most likely by renaming the underlying nodes to something such as 'PartyReference' and 'CounselorReference').

### **ECF may need to address issue with externally-defined schemas, such as digsig**

For example, the ECF schema contains a reference to a digital-signature schema defined at 'http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd'.

Initially, LexisNexis did not experience an issue with ECF including this third-party schema, but our development partners did. Their systems were sometimes unable to obtain the external schema from the designated HTTP server. This was due to the developers' or implementations' network configurations, but, revealed a little-recognized dependency in ECF for the external third-party resource, itself, to always be available.

Eventually, after the implementation was live, LexisNexis encountered another issue with the 'digsig' schema – the third-party schema includes its own embedded DTD which can cause ECF schema compilation to fail. It is not yet understood if the compilation issue was introduced to the implemented system because of some update to the underlying XML library (Microsoft's dotNet), or if the external schema was recently updated. It is also unknown if this compilation issue can be reproduced within other XML libraries offered by Sun Java, or Oracle.

LexisNexis addressed these external schema issues by adding a copy of the third-party schema to those other schema files provided with LegalXML ECF, and then editing the third-party schema and removing the embedded DTD.