

Kansas TRCC

Kansas eCitation Data Repository Design

Detailed Design Document

Version 0.3.0

Table of Contents

1	Intent	3
2	Overview	3
3	Project Vision/Scope	3
3.1	Vision	3
3.2	Scope of eCitation Data Repository	4
4	Business Context.....	4
5	Design Considerations	4
5.1	Assumptions and Dependencies.....	4
6	System Overview	4
6.1	Database	4
6.2	Repository Data Service Sub-System.....	7
6.3	NIEM XML mapping to Citations table in Repository Database	9
6.4	Reports.....	11
7	Appendix - References	12
8	Definitions	12

1 Intent

This document will define in detail the design of the eCitation Data Repository which is a SQL-server based database used for storing and indexing the eCitation records. This document does not contain the details of the various other components of the eCitation solution. The description of the overall architecture and the included components can be found in the eCitation High-level design document referenced in the Appendix.

This document is intended to serve as technical documentation of the Kansas eCitation data repository.

This is a technical document and is intended for technical audience.

2 Overview

The Traffic Records Coordinating Council (TRCC) commissioned a Strategic Plan for the development and implementation of a statewide electronic traffic citation (eCitation) system, with a central traffic citation information repository (central repository) accessible by state, local, and federal agencies, and the public. This eCitation system is an integral part of the statewide Traffic Records System (TRS) program initiated in 2005 and it will require integration with the Kansas Criminal Justice Information System (KCJIS). The TRS will be a virtual data warehouse that will provide state and local agencies with the ability to efficiently access traffic data to increase the safety of the motoring public. It will bring together information that is currently housed in separate, isolated repositories at Kansas Department of Transportation (KDOT), Kansas Highway Patrol (KHP), Kansas Department of Revenue (KDOR), Kansas Bureau of Investigation (KBI), and other agencies.

As a vital component of the TRS system, the eCitation project has been initiated with the goal of implementing a statewide eCitation system through which traffic citation data can be collected, analyzed, and distributed accurately, quickly, and cost effectively for the benefit of the public and state, local, and federal agencies.

Analysts International Corporation (AIC) has been tasked to complete a design for the production eCitation system that will serve as the foundation of the TRS 2.0 architecture. The eCitation system design includes a data repository component which will be used to store and index eCitations entered by the participating agencies.

The goal the current project phase of the eCitation Program is to provide detailed analysis and design of key eCitation components that would enhance the statewide electronic traffic citation (eCitation) prototype constructed in Phase 1B and provide guidance for implementing a hosting environment using core Microsoft technologies within the Kansas Criminal Justice Information System (KCJIS) data center to support future deployment of the eCitation system.

3 Project Vision/Scope

3.1 Vision

The goal of the statewide eCitation system is to electronically collect, analyze and distribute traffic citation data accurately, quickly and cost effectively for the benefit of the public and state, local and federal agencies.

3.2 Scope of eCitation Data Repository

The scope of this document is limited to the details of the eCitation Data Repository and does not include any details pertaining to other components utilized within the eCitation System. Thus, if some general components of eCitation (e.g. Core, Utilities) are mentioned in this document, only the details significant to the Data Repository are discussed.

The activities included within the scope of this project are as defined in the Statement of Work.

4 Business Context

The eCitation Data Repository is designed to store the key fields of an eCitation that is essential to query a record and also store the input XML message in its entirety. Every eCitation submitted to the Central repository system will be validated using a validation engine before it is stored and indexed. The goal of the eCitation program is to capture the eCitation records and to make it available for querying by the Law Enforcement stakeholders. The eCitation data repository is designed with this goal in mind.

5 Design Considerations

5.1 Assumptions and Dependencies

- The eCitation Message that are to be processed and stored in the Data Repository are expected to be in NIEM Conformant XML format.
- The eCitation Data Repository will reside in a Microsoft SQL Server database.

6 System Overview

The eCitation Data Repository is responsible for archiving and indexing citation information. The eCitation repository will expose a Web service to receive eCitation messages published by the eCitation Service discussed above. This Web service end point is intended to only receive messages from the message broker and will be designed to limit any other form of access.

The eCitation Repository will also function as the primary data source for tabular reports. The current functional requirements only call for occasional or low volume reporting and therefore it is expected that the OLTP data structures of the repository will be sufficient to support the reporting use cases. The eCitation Repository system will also act as the reporting engine backend that supports the presentation of the reporting data. This repository will also function as the primary data source for the eCitation Inquiry Service which will provide searching from outside sources.

6.1 Database

The physical database subsystem will be developed on Microsoft SQL Server 2008. This platform supports all the features required by the eCitation Repository, including structured XML data storage, full-text indexing and reporting. In particular, SQL Server Reporting Services (SSRS) functionality will act as the reporting engine for the eCitation system and provide a flexible subsystem for creating required reports for this phase or any future project phase. Utilizing this Reporting Services functionality enables delivery of reports to a client via a web browser window.

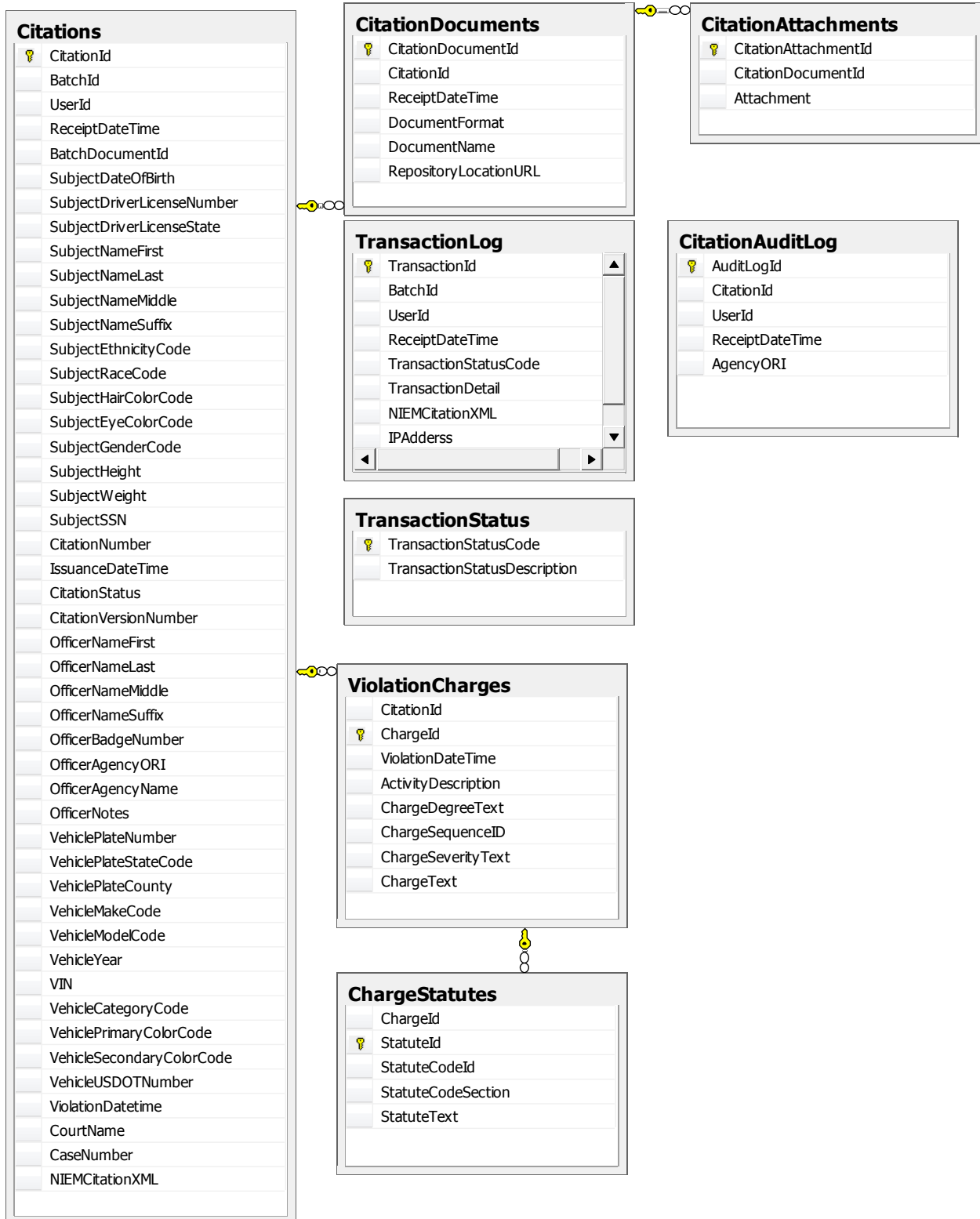
Although this database platform does support a Web service based interface to data, it would be inconvenient to use this interface to directly accept NIEM conformant eCitation XML. As a result, a Web service interface based on Microsoft Windows Communication Foundation (WCF) application layer will be developed which will in turn manage the process of persisting the information to the physical database.

The Microsoft SQL Server platform also acts as the internal repositories for the other critical subsystems, including BizTalk Server. Database administration and maintenance is simplified because all data resides on a common platform.

The Citations Database Repository consists of 8 tables

- Citations - The citations table captures submitted citations which have been given a status of either “Success” or “Warning”.
- CitationDocuments - Provides a means of storing the NIEM XML information in other formats.
- CitationAttachments - Provides the location for storing binary Filestream data from documents attached to the Citation provided by the NIEM XML.
- CitationAuditLog - Record of processed NIEM XML submitted to the Repository Data Service.
- TransactionLog - Transaction record of NIEM XML which was not successfully processed into the Citations table.
- TransactionStatus - Lookup table for valid Transaction Status in the TransactionLog table.
- ViolationCharges - Captures submitted violation/charges.
- ChargeStatutes - Captures multiple charge statutes from the violation/charges.

The physical data model of the eCitation repository is shown below.



6.2 Repository Data Service Sub-System

The primary purpose of the eCitation Repository is to store the NIEM conformant XML messages that are accepted by the eCitation Service and which have successfully been validated. The interface between the eCitation Service and the eCitation Repository was chosen to be a secure Web Service and it is called the Repository Data Service and will adhere to all security constraints outlined in the KS E-Cite Security Design documentation. The repository data service is an internal interface to be used between the BizTalk Server endpoint and will accept messages intending to be indexed and archived. This Web Service interface is not intended to be exposed to the public and will be documented here for internal use only.

Once the NIEM XML message is passed from the eCitation Service to the Repository Data Service, the service performs a few critical functions:

- Data from fields which need to be indexed are pulled out of the XML.
- Citation Image attachments are retrieved from the XML and stored in a SQL Server Database via SQL Server Filestream feature to maximize storage efficiency and retrieval performance while maintaining the original form of the document.
- The raw XML and the indexed data elements are persisted to the database for later searching and reporting.
- Appropriate audit trail entries are persisted to the database.

Should an unforeseen error occur in the processing of the NIEM XML, the error information is passed back to the eCitation Service and ultimately to the user interface. The UML diagram for the Repository Data Service components is shown in Figure 1.

Data Service Classes

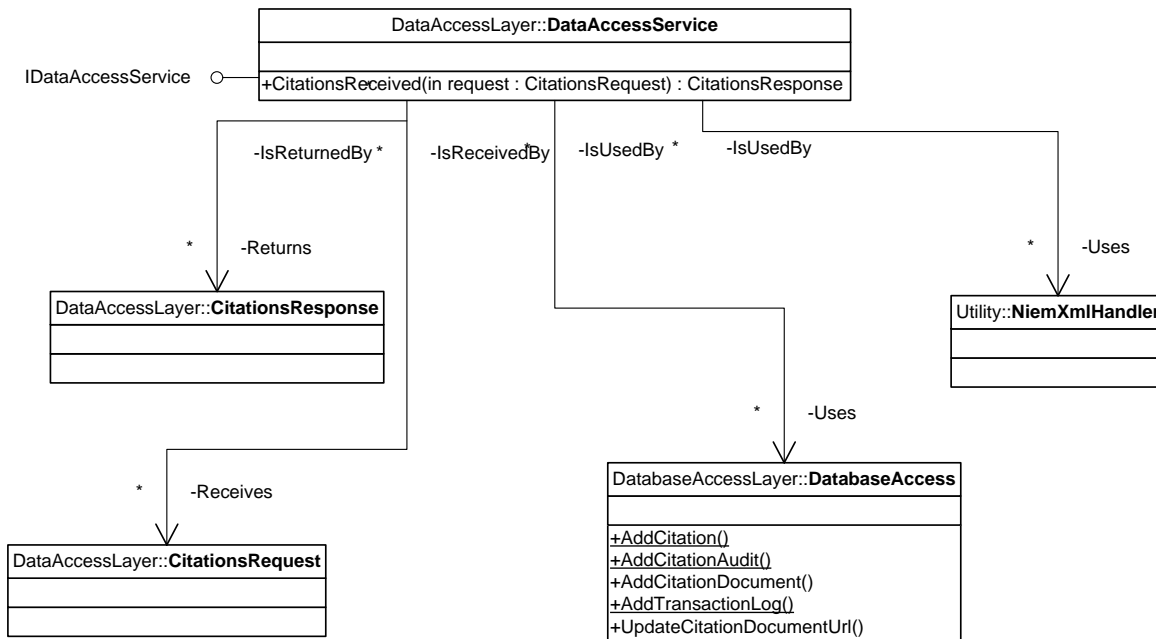


Figure 1. Repository Data Service Classes.

The Repository Data Service exposes a single method **CitationsReceived** (as shown above) for accepting NIEM XML. This **CitationsReceived** method accepts a **CitationsRequest** from the calling party.

A **CitationsRequest** consist of the following:

- **MessageContext** (Datatype: **MessageContext**). The Context of the current request which holds the **UserId** of the calling party which is recorded along with the NIEM XML and Index information in the repository.
- **TransactionStatusCode** (Datatype: **String**). Resultant status code from the Biztalk and Schematron validation process. A Status code of "Success" or "Warning" will cause the verified citation information to be entered into the Repository Database Citation table. Any other status code will cause the citation information to be recorded in just the transaction tables.
- **TransactionStatusDetail** (Datatype: **String**). Expanded resultant status code information from BizTalk
- **NiemCitations** (Datatype: **String**). NIEM xml

The **CitationsReceived** method provides a return value in the form of a **CitationsResponse**.

A **CitationsResponse** provides the following:

- **OperationStatus** (Datatype: **String**). "Success", "SystemError", or "ValidationError"
- **ErrorSource** (Datatype: **String**). Source or error if **OperationStatus** not "Success"
- **ErrorMessage** (Datatype: **String**). Full error message if **OperationStatus** not "Success"

- CitationsAdded (Datatype: integer). The number of Citations added to the Citations table in the Data Repository. This value is 0 if the initial request doesn't provide TransactionStatusCode of "Success" or "Warning".

A "Success" status in the CitationsResponse is the result of citation(s) from the NIEM XML being successfully added to the Citations table in the Data repository. A status of "SystemError" or "ValidationError" indicates a failed operation and the submitted NIEM XML data is captured in the Transaction tables. Detailed error information is then populated in the CitationsResponse back to the calling party of the Repository Data Service.

6.3 Security

The database and the repository data service interface will adhere to the security constraints outlined in the KS E-Cite Security Design documentation.

6.4 NIEM XML mapping to Citations table in Repository Database

Citations Table

Data Field	Data Type	Nullable	Origin or NIEM Xml XPath
CitationId	uniqueidentifier	no	System generated
BatchId	uniqueidentifier	no	System generated
UserId	varchar(30)	no	HttpContext UserId
ReceiptDateTime	datetime	no	Xml received datetime
BatchDocumentId	varchar(50)	yes	/ns:CitationBatchDocument/@s:id
SubjectDateOfBirth	date	yes	./nc:PersonBirthDate/nc:Date
SubjectDriverLicenseNumber	varchar(50)	yes	./j:PersonAugmentation/nc:DriverLicense/nc:DriverLicenseIdentification/nc:IdentificationID
SubjectDriverLicenseState	varchar(2)	yes	./j:PersonAugmentation/nc:DriverLicense/nc:DriverLicenseIdentification/j:DrivingJurisdictionAuthorityNCICLSTACode
SubjectNameFirst	varchar(50)	yes	./nc:PersonName/nc:PersonGivenName
SubjectNameLast	varchar(50)	yes	./nc:PersonName/nc:PersonSurName
SubjectNameMiddle	varchar(50)	yes	./nc:PersonName/nc:PersonMiddleName
SubjectNameSuffix	varchar(50)	yes	./nc:PersonName/nc:PersonNameSuffixText
SubjectEthnicityCode	varchar(50)	yes	./nc:PersonEthnicityCode
SubjectRaceCode	varchar(50)	yes	./nc:PersonRaceCode
SubjectHairColorCode	varchar(50)	yes	./nc:PersonHairColorCode
SubjectEyeColorCode	varchar(50)	yes	./nc:PersonEyeColorCode
SubjectGenderCode	varchar(50)	yes	./nc:PersonSexCode
PersonHeight	varchar(50)	yes	./nc:PersonHeightDescriptionText
PersonWeight	varchar(50)	yes	./nc:PersonWeightDescriptionText
SubjectSSN	varchar(11)	yes	./nc:PersonSSNIdentification/nc:IdentificationID
CitationNumber	varchar(50)	no	./nc:ActivityIdentification[nc:IdentificationCategoryText='CitationNumber']/nc:IdentificationID
IssuanceDateTime	datetime	no	./nc:ActivityDate/nc:DateTime
CitationStatus	varchar(50)	yes	./nc:ActivityStatus/nc:StatusText
CitationVersionNumber	int	no	./kstrs-ns:CitationVersionNumber

CourtName	varchar(50)	yes	./kstrs-ns:CitationCourtAppearance/j:CourtAppearanceCourt/j:CourtName
OfficerNameFirst	varchar(50)	yes	./nc:PersonName/nc:PersonGivenName
OfficerNameLast	varchar(50)	yes	./nc:PersonName/nc:PersonSurName
OfficerNameMiddle	varchar(50)	yes	./nc:PersonName/nc:PersonMiddleName
OfficerNameSuffix	varchar(50)	yes	./nc:PersonName/nc:PersonNameSuffixText
OfficerBadgeNumber	varchar(50)	yes	./j:CitationIssuingOfficial/j:EnforcementOfficialBadgelden- tification/nc:IdentificationID
OfficerAgencyORI	varchar(9)	no	./kstrs-ns:CitationAgency/j:OrganizationAugmentation/j:Organ- izationORIIdentification/nc:IdentificationID
OfficerAgencyName	varchar(50)	yes	./kstrs-ns:CitationAgency/nc:OrganizationName
OfficerNotes			
VehiclePlateNumber	varchar(50)	yes	./nc:ConveyanceRegistrationPlateIdentification/nc:Identifi- cationID
VehiclePlateStateCode	varchar(2)	yes	./nc:ConveyanceRegistrationPlateIdentification/j:Identific- ationJurisdictionNCICLISCode
VehiclePlateCounty	varchar(50)	yes	./nc:ConveyanceRegistrationPlateIdentification/nc:Identifi- cationJurisdictionText
VehicleMakeCode	varchar(50)	yes	./nc:VehicleMakeCode
VehicleModelCode	varchar(50)	yes	./nc:VehicleModelCode
VehicleYear	varchar(50)	yes	./nc:ItemFirstSoldYearDate
VIN	varchar(50)	yes	./nc:VehicleIdentification/nc:IdentificationID
VehicleCategoryCode	varchar(50)	yes	./kstrs-ns:CitationVehicle/j:VehicleBodyCategoryCode
VehiclePrimaryColorCode	varchar(50)	yes	./nc:VehicleColorPrimaryCode
VehicleSecondaryColorCode	varchar(50)	yes	./nc:VehicleColorSecondaryCode
VehicleUSDOTNumber	varchar(50)	yes	./kstrs-ns:CitationVehicle/j:CommercialCarrierUSDOTNumber
ViolationDatetime	datetime	yes	./kstrs-ns:CitationViolation/nc:ActivityDate/nc:DateTime
CaseNumber	varchar(50)	yes	./kstrs-ns:CitationCaseIdentification/nc:IdentificationID
NIEMCitationXML	xml	no	

ViolationCharges Table

Data Field	Data Type	Nullable	Origin or NIEM Xml XPath
CitationId	uniqueidentifier	no	System generated
ChargeId	uniqueidentifier	no	System generated
ViolationDateTime	datetime	yes	./kstrs-ns:CitationViolation/nc:ActivityDate/nc:DateTime
ActivityDescription	varchar(50)	yes	./kstrs-ns:CitationViolation/kstrs-ns:ViolationCharge/nc:ActivityDescriptionText
ChargeDegreeText	varchar(50)	yes	./kstrs-ns:CitationViolation/kstrs-ns:ViolationCharge/j:ChargeDegreeText
ChargeSequenceID	varchar(50)	yes	./kstrs-ns:CitationViolation/kstrs-ns:ViolationCharge/j:ChargeSequenceID
ChargeSeverityText			./kstrs-ns:CitationViolation/kstrs-ns:ViolationCharge/j:ChargeSeverityText
ChargeText			./kstrs-ns:CitationViolation/

ChargeStatutes Table

Data Field	Data Type	Nullable	Origin or NIEM Xml XPath
ChargeId	uniqueidentifier	no	System generated
StatuteId	uniqueidentifier	no	System generated
StatuteCodeId	Varchar(50)	yes	./kstrs-ns:CitationViolation/kstrs-ns:ViolationCharge/j:ChargeStatute/j:StatuteCodeIdentification/nc:IdentificationID
StatuteCodeSection	Varchar(50)	yes	./kstrs-ns:CitationViolation/kstrs-ns:ViolationCharge/j:ChargeStatute/j:StatuteCodeSectionIdentification/nc:IdentificationID
StatuteText	Varchar(50)	yes	./kstrs-ns:CitationViolation/kstrs-ns:ViolationCharge/j:ChargeStatute/j:StatuteText

CitationAttachments Table

Data Field	Data Type	Nullable	Origin or NIEM Xml XPath
CitationAttachmentId	uniqueidentifier	no	System generated
CitationDocumentId	uniqueidentifier	no	System generated
Attachment	filestream	no	./kstrs-ns:CitationImage/nc:BinaryBase64Object

6.5 Reports

SQL Server Reporting Services provides a report based on data in the eCitation Repository Database. The initial report included, developed during the Prototype Phase, with the eCitation Repository Database is the “CITATION PROCESSING STATUS SUMMARY REPORT”. This report displays submitted NIEM XML along with some index fields and metadata captured during the submission of the NIEM XML to the repository. This report is intended for development/support use and is not a report for an end user. The following fields are display in this report:

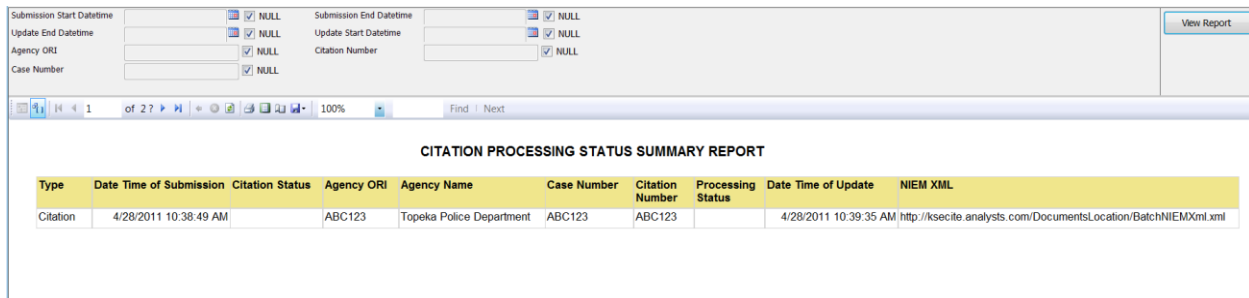
- Citation Type
- Date Time of Submission
- Citation Status
- Agency ORI
- Agency Name
- Case Number
- Citation Number
- Processing Status
- Date Time of Update
- NIEM XML link

These report fields are generated from a SQL Server stored procedure which collects submitted citation data from captured NIEM XML and supplies the report with a summary of citation data in the repository.

Report results can be filtered by entering data for the report parameters. Any of these parameters can be excluded from the report by passing a null value for the associated parameter. The following report parameters are available in this report:

- Submission Start DateTime
- Submission End DateTime
- Update Start DateTime
- Update End DateTime
- Agency ORI
- Citation Number
- Case Number

SQL Server Reporting Services provides the interface for entering report criteria, viewing, and printing reports. This reports interface is depicted below.



Type	Date Time of Submission	Citation Status	Agency ORI	Agency Name	Case Number	Citation Number	Processing Status	Date Time of Update	NIEM XML
Citation	4/28/2011 10:38:49 AM		ABC123	Topeka Police Department	ABC123	ABC123		4/28/2011 10:39:35 AM	http://ksecite.analysts.com/Documents/Location/BatchNIEMxml.xml

7 Appendix - References

Kansas eCitation IEPD	http://www.kstrs.org/Traffic%20Reports/XML%20Schemas/KS-Citation-IEPD-v2.0.3.zip
Kansas eCitation High-level Design Document	https://projects.analysts.com/kansasecitation/internal/Shared%20Documents/Phase%201B%20Deliverables/KS%20E-Cite%20High-level%20Design%20V1.2.docx
Kansas eCitation Data Requirements	http://www.kstrs.org/Traffic%20Reports/eCitation/eCitation%20Data%20Requirements-v1.1.docx
SOW	https://projects.analysts.com/kansasecitation/SignedDocuments/Kansas%20eCitation%20Project%20Phase%201C%20SOW%2012.28.2010.pdf
KS E-Cite Security Design Document	https://projects.analysts.com/kansasecitation/TechnicalDocuments/Security%20Model/KS%20E-Cite%20Security%20Detailed%20Design%20v1.0.0.docx

8 Definitions

Acronym/Term	Definition
--------------	------------

NIEM	National Information Exchange Model
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language
XSLT	eXtensible Stylesheet Language Transformations
XPath	XML Path Language
ISO	International Organization for Standardization