



## Uniform Crime Reporting (UCR)

# UCR Interface Control Document with Web Services

Version 1.0

03/22/2018



CJIS Document Number – UCR-DOC-02509-1.0

Prepared by:

Criminal Justice Information Services Division  
Information Technology Management Section (ITMS)  
Law Enforcement Technical Support Unit (LETSU)

# TABLE OF CONTENTS

<b>1</b>	<b>Scope</b> .....	<b>1</b>
1.1	System Identification.....	1
1.1.1	National Incident-Based Reporting System (NIBRS) Services System.....	1
1.1.2	Service Clients System(s).....	1
1.2	Document Overview.....	1
<b>2</b>	<b>Description</b> .....	<b>2</b>
2.1	System Overview.....	2
2.2	Interface Overview.....	2
2.3	Functional Allocation.....	2
2.4	Transactions.....	3
2.4.1	Basic NIBRS Reporting Data Submission Workflow.....	3
2.4.2	“Warning” and “Error” Querying.....	3
2.4.3	Resubmission of NIBRS Data.....	4
2.4.4	Supporting Services.....	4
2.5	Security and Integrity.....	5
<b>3</b>	<b>Detailed Interface Requirements</b> .....	<b>5</b>
3.1	Interface 1 Requirements.....	5
3.1.1	Message (or File) Requirements.....	5
3.1.2	Interface Initiation.....	12
3.1.3	Flow Control.....	12
3.1.4	Security Requirements.....	12

# 1 SCOPE

---

This document provides an outline for use in the specification of requirements imposed on one or more systems, subsystems, Hardware Configuration Items, Computer Software Configuration Items, manual operations, or other system components to achieve one or more interfaces among these entities. An Interface Control Document (ICD) can cover requirements for any number of interfaces to a system.

## 1.1 System Identification

The following subsections shall contain a full identification of the systems participating in the interface, the interfacing entities, and the interfaces to which this document applies. This should include, as applicable, identification numbers(s), title(s), abbreviation(s), version number(s), release number(s), or any lower level version descriptors used.

### 1.1.1 National Incident-Based Reporting System (NIBRS) Services System

The National Incident-Based Reporting System (NIBRS) Services provide for the full interactions between law enforcement submitters, state program office repositories, and the FBI's UCR Program.

The process of reporting incidents to the FBI's UCR program involves four sets of actors:

1. Law enforcement agencies that originate incident-based reports;
2. Federal agencies that originate incident-based reports
3. State UCR programs that consolidate incident reports from their constituent reporting agencies, often in a statewide repository;
4. And the FBI-CJIS Division's UCR Program.

The NIBRS Services processes are bi-directional, encompassing response messages flowing back from state programs and UCR.

### 1.1.2 Service Clients System(s)

A service-oriented view of the NIBRS Services consider each service client as a provider and consumer of data. The following description outlines the full set of services:

- Law enforcement agencies submitting incident records to the FBI;
- A State Program transmitting records from the repository to the FBI;
- The return of acknowledgements, status, and errors from both processing systems (state repositories and the FBI); and finally,
- The re-submission to correct errors reported by either of the submitting systems.

## 1.2 Document Overview

This document will reflect the CJIS Division's specifications for web service communication via the NIBRS Services. Technical implementation details of specific interfaces with client systems shall be detailed in separate documentation.

## 2 DESCRIPTION

---

The purpose of the NIBRS Services is to foster machine-to-machine communication for the complex, multi-stage, round-trip business process of NIBRS submissions. Services will shorten the communication cycle from an agency client to the FBI UCR System, ultimately making validated crime statistical information accessible to practitioners, researchers and the public closer to the time the crime has occurred.

These services are also designed to improve the quality of the information in the UCR repository. A service-oriented exchange will allow more real-time validation, done closer to the point of origin, and streamline the re-submission of reports redress errant data.

### 2.1 System Overview

The FBI UCR system hosts the NIBRS Services. NIBRS data service clients utilize the processes to submit NIBRS data and query for data status; e.g. data that has validated via the FBI UCR validation process.

NIBRS data is submitted to the NIBRS Services via compliant National Information Exchange Model (NIEM) Information Exchange Package Documentation (IEPD) documents. The data specification version is subject to change in partnership with the provider-consumer community.

Service clients can also query the NIBRS Services in order to ascertain status of submitted data within the FBI UCR validation process. This process returns specific information related to the submitted data in statuses of “Warning,” “Error,” “Processing” or “Accepted.”

Details of “Warning” and “Error” coded data can be found within the NIBRS Technical Specification document distributed by the FBI UCR program.

### 2.2 Interface Overview

Service clients interface with the NIBRS Services via the public internet using the FBI CJIS Division’s services.cjis.gov address. The full address is included in the Web Service Description Language (WSDL) file distributed to vetted providers and-or consumers by the FBI UCR program.

The provider and-or consumer can submit NIBRS data via the service endpoints as well as query the status of NIBRS data in order to validate submissions. Data provided to the NIBRS Services is processed through the FBI UCR validation engine.

### 2.3 Functional Allocation

The NIBRS Services are comprised of eight processes. Each process interacts with the FBI UCR system in order to either ingest, alter, or query for data from the repository.

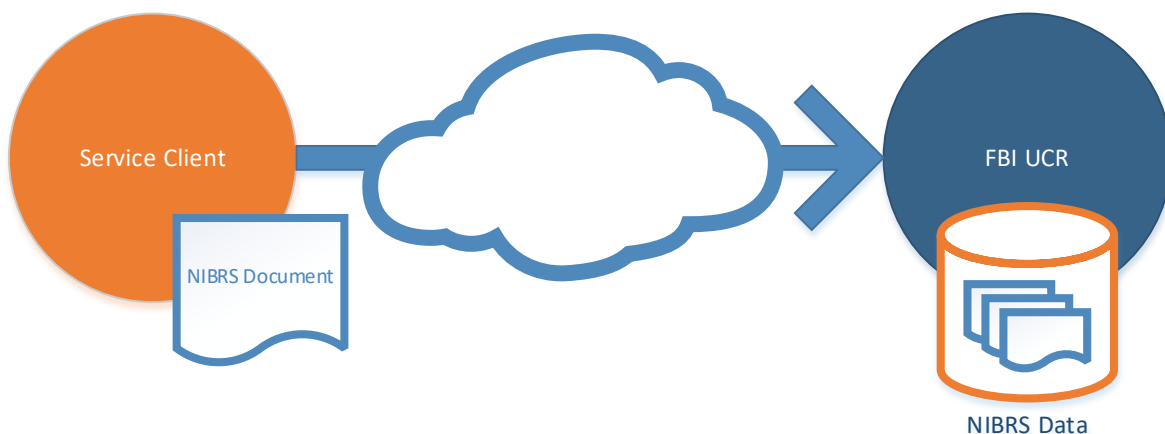
NIBRS Service Process	Description
SubmitNibrsNIEMDocument	This process accepts NIBRS data via a NIEM IEPD formatted document. It returns a document name that can be used in other processes to query status.
CheckStatusByDDocname	This process is used to query the status of NIBRS data by document name.

ListViolationsByDDocname	This process returns a listing of <i>active</i> “Warnings” and-or “Errors” codes associated with a document name.
ListViolationsByORIAndDateRange	This process scopes ListViolationsByOri by a date range.
ListViolationsByORIByYear	This process scopes ListViolationsByOri by year.
ListUploadsByORIAndDateRange	This process scopes ListUploadsByORI by a date range.
ClearWarning	This process is used to update the status of <i>a specific</i> “Warning” code of a NIBRS document to “inactive.”
ClearAllWarnings	This is process used to update the status of <i>all</i> “Warning” codes of a NIBRS document to “inactive.”

## 2.4 Transactions

### 2.4.1 Basic NIBRS Reporting Data Submission Workflow

Service clients utilize the SubmitNibrsNIEMDocument process to submit NIEM IEPD documents to the FBI UCR repository. Each document is initially validated using XSD Schema validation. Once a document successfully passes schema validation, a document is then validated using the FBI UCR NIBRS business rules and stored. The transaction returns a document name (ddocname) that is comprised of the date of the incident, incident number and NCIC ORI as included within the document, the final status of the document (ACCEPTED, WARNINGS, ERRORS), any violation(s) for the document, and the action type associated with the document.

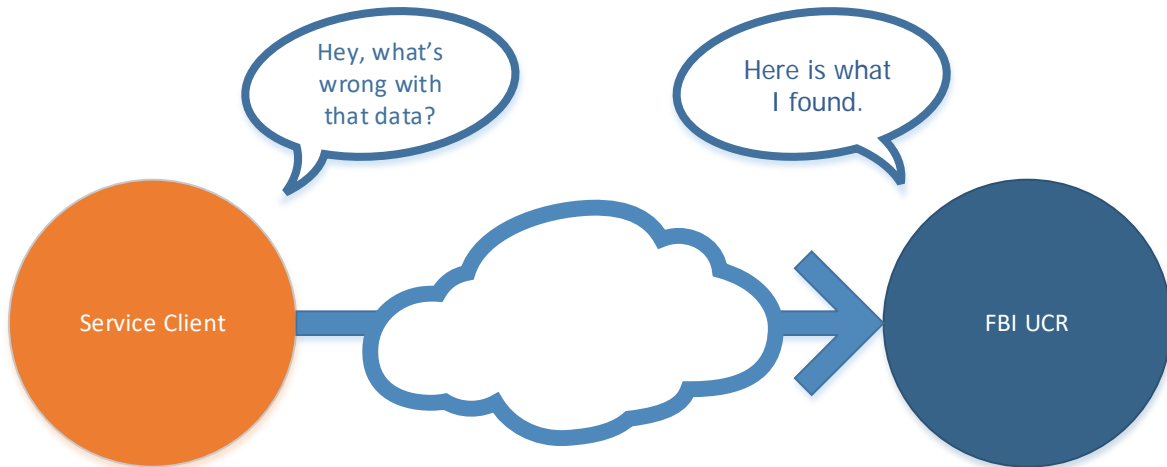


### 2.4.2 “Warning” and “Error” Querying

If a document is in a violation status, a service client can query the NIBRS Services for a listing of those violations using the ListViolations processes. Violations are either WARNING(s) or ERROR(s). These are also returned on the initial submission of the document to the NIBRS Services.

Document WARNING(s) can be “cleared” using ClearWarning or ClearAllWarning processes.

Document ERROR(s) can’t be “cleared” and the document must be resubmitted with the specified data corrected.



### 2.4.3 Resubmission of NIBRS Data

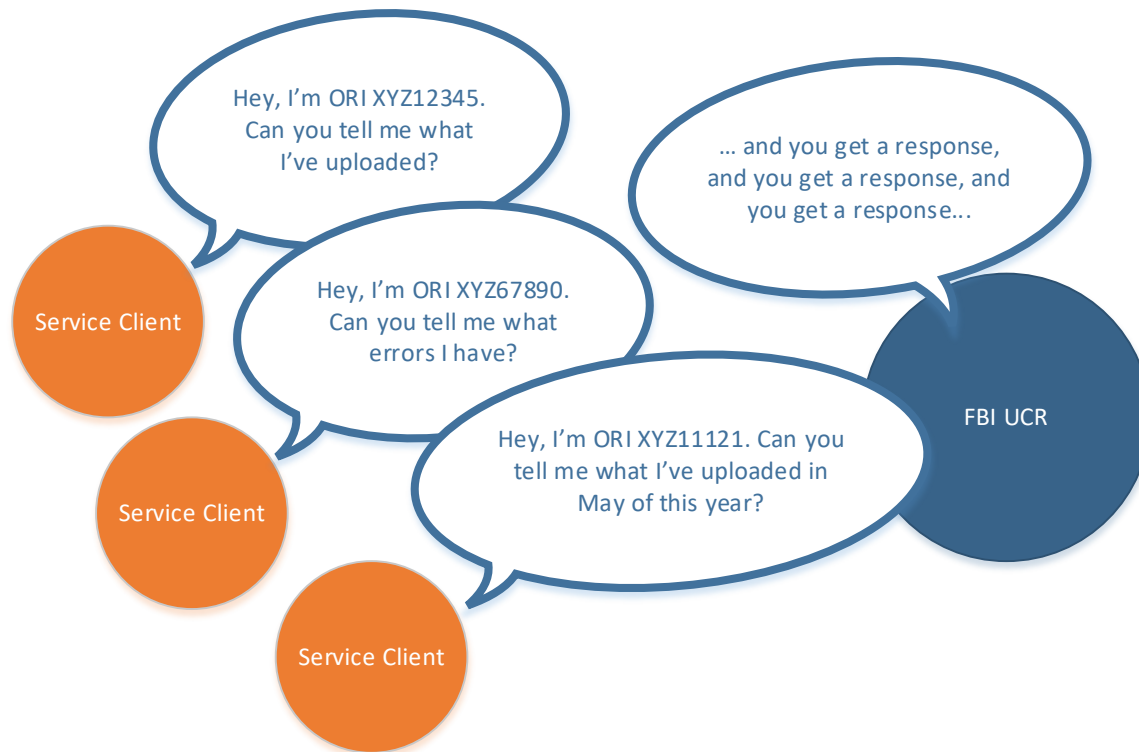
In order to resubmit NIBRS data, the service client will need to send the corrected NIBRS IEPD data with the appropriate ReportActionCategoryCode set to “R” for replace, or “D” to delete the NIBRS data on file, and then “I” for incident report or “A” for Group B/Zero report to insert it again.

*NOTE: Please refer to the NIBRS Technical Specification documentation available from the FBI UCR program.*



### 2.4.4 Supporting Services

A service client may utilize an ORI to query for data historically submitted to the FBI UCR system using the ListUploadsByORI\* processes. These processes return all historic data for that ORI unless a scoped query is used to narrow the date range. The data that is returned is based on all submissions that have been made to the New UCR System, not legacy data that was migrated.



## 2.5 Security and Integrity

For authentication and encryption purposes, service clients will need to provide a self-signed or purchased certificate to the FBI UCR program. Transaction data will be encrypted from the service client to the FBI UCR system, however, the data is stored unencrypted.

This process is defined in a separate document provided by the FBI UCR Program Office.

## 3 DETAILED INTERFACE REQUIREMENTS

---

This section will seek to provide more detailed descriptions for utilizing the NBIRS Services. Where applicable, data standards should be deferred to their respective current documentation.

### 3.1 Interface 1 Requirements

All transactions with the NIBRS Services are handled via Simple Object Access Protocol (SOAP) over the public internet. The messages are encrypted between the client and service.

#### 3.1.1 Message (or File) Requirements

NIBRS report submissions must be structured to the NIEM IEPD markup standard. The NIEM IEPD documentation is available from the FBI UCR program.

### 3.1.1.1 Basic NIBRS Reporting Data Submission Workflow

#### 3.1.1.1.1 SubmitNibrsNIEMDocument Data Assembly Characteristics

```
<soapenv:Envelope>
  <soapenv:Header/>
  <soapenv:Body>
    <ws:SubmitNibrsNIEMDocument>
      <xmlDoc><![CDATA[INSERT NIEM IEPD XML HERE]]></xmlDoc>
    </ws:SubmitNibrsNIEMDocument>
  </soapenv:Body>
</soapenv:Envelope>
```

#### 3.1.1.1.2 SubmitNibrsNIEMDocument Response Characteristics

```
<S:Envelope>
  <S:Body>
    <ns2:SubmitNibrsNIEMDocumentResponse>
      <return>
        <ingestResponse>
          <createdDate>[YYYYMMDD]</createdDate>
          <ddocname>[Document Name]</ddocname>
          <violations>
            <badValue >[...]</badValue>
            <dataElement>[...]</dataElement>
            <propertyLossCode>[...]</propertyLossCode>
            <segmentLevel>[...]</segmentLevel>
            <sequenceNumber>[...]</sequenceNumber>
            <violationCode >[...]</violationCode>
            <violationDescription >[...]</violationDescription>
            <violationLevel >[...]</violationLevel>
            <typePropLoss>[...]</typePropLoss>
          </violations>
          <status>ERRORS</status>
          <actionType>I</actionType>
        </ingestResponse>
      </return>
    </ns2: SubmitNibrsNIEMDocumentResponse>
  </S:Body>
```



```
</S:Envelope>
```

### 3.1.1.1.3 CheckStatusByDDocname Data Assembly Characteristics

```
<soapenv:Envelope>  
  <soapenv:Header/>  
  <soapenv:Body>  
    <ws:CheckStatusByDDocname>  
      <ori>ORI</ori>  
      <ddocname>Document Name</ddocname>  
    </ws:CheckStatusByDDocname>  
  </soapenv:Body>  
</soapenv:Envelope>
```

### 3.1.1.1.4 CheckStatusByDDocname Response Characteristics

```
<S:Envelope>  
  <S:Body>  
    <ns2:CheckStatusByDDocnameResponse>  
      <return>  
        <ddoc>  
          <createdDate>YYYYMMDD</createdDate>  
          <ddocname>Document Name</ddocname>  
          <status>Status</status>  
          <actionType>actionType</actionType>  
        </ddoc>  
      </return>  
    </ns2:CheckStatusByDDocnameResponse>  
  </S:Body>  
</S:Envelope>
```

## 3.1.1.2 DDocname “WARNINGS” and “ERRORS” Querying

### 3.1.1.2.1 ListViolationsByDDocname Data Assembly Characteristics

```
<soapenv:Envelope>  
  <soapenv:Header/>  
  <soapenv:Body>  
    <ws:ListViolationsByDDocname>  
      <ori>ORI</ori>  
      <ddocname>Document Name</ddocname>
```

```
</ws:ListViolationsByDDocname>
</soapenv:Body>
</soapenv:Envelope>
```

### 3.1.1.2.2 ListViolationsByDDocname Response Characteristics

```
<S:Envelope>
  <S:Body>
    <ns2:ListViolationsByDDocnameResponse>
      <return>
        <ddoc>
          <!--See 3.1.1.4.4 Supporting Services Response Characteristics
-->
          </ddoc>
        </return>
      </ns2:ListViolationsByDDocnameResponse>
    </S:Body>
  </S:Envelope>
```

### 3.1.1.3 Clearing “WARNINGS”

#### 3.1.1.3.1 ClearWarning Data Assembly Characteristics

```
<soapenv:Envelope>
  <soapenv:Header/>
  <soapenv:Body>
    <ws:ClearWarning>
      <ori>ORI</ori>
      <ddocname>Document Name</ddocname>
      <warningCode>Code</warningCode>
    </ws:ClearWarning>
  </soapenv:Body>
</soapenv:Envelope>
```

#### 3.1.1.3.2 ClearWarning Response Characteristics

```
<S:Envelope>
  <S:Body>
    <ns2:ClearWarningResponse>
      <return>
        <objectsUpdated>[...]</objectsUpdated>
```

```

    <success>Boolean</success>
  </return>
</ns2:ClearWarningResponse>
</S:Body>
</S:Envelope>

```

### 3.1.1.3.3 ClearAllWarnings Data Assembly Characteristics

```

<soapenv:Envelope>
  <soapenv:Header/>
  <soapenv:Body>
    <ws:ClearAllWarnings>
      <ori>ORI</ori>
      <ddocname>Document Name</ddocname>
    </ws:ClearAllWarnings>
  </soapenv:Body>
</soapenv:Envelope>

```

### 3.1.1.3.4 ClearAllWarnings Response Characteristics

```

<S:Envelope>
  <S:Body>
    <ns2:ClearAllWarningsResponse>
      <return>
        <objectsUpdated>...</objectsUpdated>
        <success>Boolean</success>
      </return>
    </ns2:ClearAllWarningsResponse>
  </S:Body>
</S:Envelope>

```

## 3.1.1.4 Supporting Services

### 3.1.1.4.1 ListUploadsByORIAndDateRange Data Assembly Characteristics

```

<soapenv:Envelope>
  <soapenv:Header/>
  <soapenv:Body>
    <ws:ListUploadsByORIAndDateRange>
      <ori>[ORI]</ori>
      <startDate>YYYYMMDD</startDate>
    </ws:ListUploadsByORIAndDateRange>
  </soapenv:Body>
</soapenv:Envelope>

```

```

    <endDate>YYYYMMDD</endDate>
  </ws:ListUploadsByORIAndDateRange>
</soapenv:Body>
</soapenv:Envelope>

```

#### 3.1.1.4.2 ListViolationsByORIAndDateRange Data Assembly Characteristics

```

<soapenv:Envelope>
  <soapenv:Header/>
  <soapenv:Body>
    <ws:ListViolationsByORIAndDateRange>
      <ori>ORI</ori>
      <startDate>YYYYMMDD</startDate>
      <endDate>YYYYMMDD</endDate>
    </ws:ListViolationsByORIAndDateRange>
  </soapenv:Body>
</soapenv:Envelope>

```

#### 3.1.1.4.3 ListViolationsByORIByYear Data Assembly Characteristics

```

<soapenv:Envelope>
  <soapenv:Header/>
  <soapenv:Body>
    <ws:ListViolationsByORIByYear>
      <ori>ORI</ori>
      <year>YYYY</year>
    </ws:ListViolationsByORIByYear>
  </soapenv:Body>
</soapenv:Envelope>

```

#### 3.1.1.4.4 Supporting Services Response Characteristics

```

<S:Envelope>
  <S:Body>
    <ns2:ListViolationsByORIByYearResponse>
      <return>
        <ddoc>
          <createdDate>YYYYMMDD</createdDate>
          <ddocname>Document Name</ddocname>
          <violations>

```

```

        <badValue >...</badValue>
        <dataElement>...</dataElement>
        <propertyLossCode>...</propertyLossCode>
        <segmentLevel>...</segmentLevel>
        <sequenceNumber>...</sequenceNumber>
        <violationCode >...</violationCode>
        <violationDescription >...</violationDescription>
        <violationLevel >...</violationLevel>
        <typePropLoss>...</typePropLoss>
    </violations>
    <status>ERRORS</status>
    <actionType>I</actionType>

</ddoc>
<ddoc>
    <createdDate>[YYYYMMDD]</createdDate>
    <ddocname>[Document Name]</ddocname>
    <violations>
        <badValue >...</badValue>
        <dataElement>...</dataElement>
        <propertyLossCode>...</propertyLossCode>
        <segmentLevel>...</segmentLevel>
        <sequenceNumber>...</sequenceNumber>
        <violationCode >...</violationCode>
        <violationDescription >...</violationDescription>
        <violationLevel >...</violationLevel>
        <typePropLoss>...</typePropLoss>
    </violations>
    <status>WARNINGS</status>
    <actionType>I</actionType>
</ddoc>
</return>
</ns2:ListViolationsByORIByYearResponse>
</S:Body>
</S:Envelope>

```

### 3.1.2 Interface Initiation

The NIBRS Services are available 24/7 365 days a year, barring scheduled/unforeseen maintenance.

Submitting agency contacts will be notified by UCR Operations staff at least 72 hours in advance of scheduled maintenance. In the event that there needs to be unscheduled maintenance due to unforeseen circumstances, all submitting agency contacts will be notified as quickly as possible.

Submitting agency contacts will be notified via the email addresses that are currently on file.

### 3.1.3 Flow Control

Inbound data validation occurs different levels depending on the NIBRS Service process.

The NIBRS Service will validate inbound NIEM IEPD submissions specific to the published IEPD standard available from the FBI UCR program.

Additionally, the NIBRS Services expects valid data when querying supporting processes. For example, when querying for violations via ORI, valid ORI data of alpha-numeric data must be used. When using additional query data such as dates, they must conform to the expected format. Here's example of a set of "queryErrors" in a return.

```
<queryErrors>
  <error>startDate cannot be empty and must be valid dates</error>
  <error>endDate cannot be empty and must be valid dates</error>
  <error>ori cannot be empty and must be an alphanumeric</error>
</queryErrors>
```

In the event of a FBI UCR system failure, the service client may receive an error with a message stating to contact UCR.

### 3.1.4 Security Requirements

In order to establish connectivity to the NIBRS Services a service client will need to provide an identity certificate and a static IP or subnet to the FBI UCR program. This certificate will be utilized to authenticate the service client to the NIBRS Service. The program will accept self-signed certificates at the time of this document's writing.

The CJIS Division requires support of TLS 1.2 encryption for transactions to the NIBRS Services. TLS 1.2 is defined in RFC 5246 (<https://tools.ietf.org/html/rfc5246>).