NEXTSTEP API Getting Started Guide

This document will serve as a guide to understanding the fundamentals involved in integrating applications with Concord's NEXTSTEP platform. NEXTSTEP provides a simple yet powerful way to connect documents and the data they contain with workflow processes and critical applications. Using NEXTSTEP APIs, you can integrate data capture and workflow processes with fax documents as well as documents that originate from other sources. The NEXTSTEP Intelligent Capture service can be applied to documents to extract meaningful data and deliver that information to the applications that house and manage your documents and data.

All public APIs exposed by NEXTSTEP use Representational State Transfer (REST) architecture. There are two primary categories for NEXTSTEP API processing: **Document Intake** and **Document Return.**

Document Intake provides a method to inject documents and data into the NEXTSTEP platform. Documents that are being received and processed via fax do not require a customer to utilize Document Intake, as this is managed by Concord. Where other document sources are relevant, Document Intake may be used to inject documents into the NEXTSTEP platform for processing.

Document Return provides a method to deliver documents and/or extracted data to applications that customers host. This may be the result of processing an inbound fax or may be the result of a document that is injected via Document Intake. In either case, the Document Return method is used to deliver the processed document and extracted data to a customer defined endpoint.

Both APIs may be relevant to your needs, depending on the source of the documents that are intended to be processed by NEXTSTEP. Documents that are injected into NEXTSTEP may be delivered to two possible destinations: The **NEXTSTEP UI**, where they can be accessed by users operating in NEXTSTEP's portal based shared queues, and/or your application itself - via the **Document Return API**, which is used to transmit data directly to customer systems.

To leverage the NEXTSTEP APIs, a customer will need to request a NEXTSTEP account via their relationship manager. Concord will provide an API key that will be used to authenticate the requests being made to Concord when Intake submissions are made.

Document Intake

Document Intake is the process by which new documents are submitted to the NEXSTEP platform. For the purposes of this document, we will be covering the "Generic" Document Intake, which allows an application to submit documents that do not originate as a fax message. Details regarding the Generic Document Intake method may be found here:

https://developer.concordfax.com/nextstep/html/d7c22cb4-f719-4acd-b0ac-7cdbf6a88f21.htm

Intake supports submission of documents in the following formats:

- Tagged Image File Format (TIFF)
- Joint Photographic Experts Group Format (JPEG)
- Portable Network Graphics Format (PNG)
- Portable Document Format (PDF)

Using Document Intake, documents can be submitted to a **Process** or a **Queue**, or to both if the situation warrants this.

Process: A NEXTSTEP Process defines a series of activities designed to prepare and transform an incoming document into the central artifact in a business process. Processing may include data extraction, barcode scanning, signature detection or conversion to searchable PDF. Submission to a Process is used when there is no transfer of data to the NEXTSTEP UI, and documents are processed via NEXTSTEP to extract data that will be transferred directly to a customer application via the Document Return API.

Queue: A NEXTSTEP Queue is virtual location that is accessible via the NEXTSTEP UI interface. Documents that are submitted to a NEXTSTEP Queue are made available to UI users who are enabled for access to that Queue. NEXTSTEP may be used to scan documents for data extraction, barcode scanning, signature detection or conversion to searchable PDF, as with Processes, but the resulting data is displayed in the NEXTSTEP UI via a Queue, rather than submitted directly to a customer endpoint.

Document Processing

A customer may indicate to Concord what NEXTSTEP processing should be performed when the document is submitted to the NEXTSTEP Process or Queue. Concord will configure the appropriate actions to be taken against the document from the following set of current possibilities, which will expand over time:

- Intelligent Data Extraction of 1 or more specific fields (Currently: Patient Name, Date of Birth, Date of Encounter, Social Security Number, Medical Record Number)
- Barcode Reading
- Signature Detection
- Full-page OCR extraction
- Searchable PDF conversion
- Document Classification

The results of the processing performed on the document will then be delivered to the customer via API (Document Return) or to the Queue designed for UI availability.

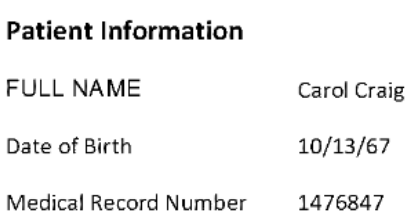
Properties

Documents submitted to a Process and/or Queue may also contain **Properties, a** set of name-value pairs that define additional metadata that should be stored with the document. When documents are submitted via Fax, NEXTSTEP captures all of the metadata about the fax transmission and makes it available to the document. When documents are submitted by third party applications, those applications have the ability to submit and associate metadata specific to their use case via **Properties.** Properties may be used in conjunction with Queues, and customers may include data that can be represented to users operating in the UI, to provide identification and indexing elements associated with that document.

Below is example HTTP POST request for the intake API (in raw text format) is provided below, showing submission to both a Process and a Queue with name-value properties included:

```
POST /api/intake HTTP/1.1
Host: https://nextstep.concord.net
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundary7MA4YWxkTrZu0gW
X-Auth-User: 12345
X-Auth-ApiKey: 12345-6789-1212-12345-0001

Content-Disposition: form-data; name="metadata"

{
  "sourceId": "bbe58db7-b3f3-461a-9c47-5b803586cc53",
  "process": {
    "accountId": "1234567890",
    "processId": "1234567891"
  },
  "queue": {
    "accountId": "1234567890",
    "queueId": "1234567892"
  },
  "properties": {
    "Patient Name": "Carol Craig",
    "Date of Birth": "10/13/67",
    "MRN": "1476847"
  }
}

Content-Disposition: form-data; name="file"; filename="undefined"
Content-Type: file

<binary file data>

------WebKitFormBoundary7MA4YWxkTrZu0gW--
```

Data submitted as "Properties" to a Queue will be displayed as Custom Fields in the NEXTSTEP UI:



Document Return

Document Return is the process via which NEXTSTEP delivers documents and meta-data content to a customer. For the purposes of this document, we will be focusing on Document Return via HTTPS Push:

https://developer.concordfax.com/nextstep/html/6b7f7e5a-8f16-4ce0-bd38-684dc4446e54.htm

(NOTE: Document Return via Pull is also available: https://developer.concordfax.com/nextstep/html/455d09e3-1c59-4d64-a62a-e4266a95ca08.htm)

Document Return via HTTP PUSH requires a customer to host a web endpoint that can receive the document image and metadata files from the NEXTSTEP platform. The files and format of the meta-data that will be included in the delivery request will be defined by the process configuration, which will be documented prior to API integration.

The PUSH delivery request may come as a result of processing related to a document submitted via **Document Intake** or may be the result of an inbound fax that has been processed by NEXTSTEP to extract specific data for transfer to customer's endpoint. The data transfer from NEXTSTEP occurs immediately after NEXTSTEP processing completes and is not batched. Each processed document will be relayed as a single HTTPS transmission, containing both document and meta-data.

Requests from the platform for document delivery will always be an HTTP POST with a multipart form data content type, containing the headers and content sections described below:

Headers & Form Content Sections

Name	Location	Sample Value & Description
Delivery URL	Configuration	*https://api.your-domain.com/api/nextstep/delivery* During discovery and configuration of your NEXTSTEP process, you will need to provide the host name where delivery requests should be sent. *The values above are for example purposes only*
Authorization	Header	*Bearer mF_9.B5f-4.1JqM* The authorization header will be populated with the static bearer token provided to the platform during configuration. (Optional) Currently, only bearer token authentication is supported.
<File Name>	Content Sections	*<binary file data>* Each configured delivery file will be added to the request in a separate content section. Each section will include the customer specified name to represent the file, as well as a logical file name, with file type extension,

Name	Location	Sample Value & Description
		and content type header value. The bytes representing the file content will be sent as the section body.

Below is a sample request in JSON format that posts both an image file and metadata file to the delivery endpoint. In this example the data did not originate from a fax and there is no accompanying fax transmission data:

```
POST /api/nextstep/delivery
Host: https://api.your-domain.com/
Content-Type: multipart/form-data; boundary=----
WebKitFormBoundaryWfPNVh4wuWBlyEyQ

------WebKitFormBoundaryWfPNVh4wuWBlyEyQ

Content-Disposition: form-data; name="file"; filename="123456789.tif"
Content-Type: image/tiff

<binary file data>

------WebKitFormBoundaryWfPNVh4wuWBlyEyQ

Content-Disposition: form-data; name="metadata"; filename="metadata.json"
Content-Type: application/json

{
  "sourceId": "123456789",
  "fields": [
    {
      "name": "patient_name",
      "value": "John Smith",
      "confidence": 0.8
    },
    {
      "name": "patient_dob",
      "value": "02/20/1964",
      "confidence": 1.0
    },
    {
      "name": "date_of_encounter",
      "value": "01/30/2019",
      "confidence": 0.95
    },
    {
      "name": "mrn",
      "value": "142637A",
      "confidence": 0.9
    }
  ]
```

```
}

------WebKitFormBoundaryWfPNVh4wuWBlyEyQ
```

Below is a sample request for a document received via fax in XML format that includes inbound fax data as well as data related to content extraction:

```
<Document>
  <ExtractionResult>
    <ExtractionPage>
      <PageNumber>1</PageNumber>
      <ExtractionField>
        <Name>patient_name</Name>
        <Confidence>0.800000011920929</Confidence>
        <Value>John Smith</Value>
      </ExtractionField>
      <ExtractionField>
        <Name>patient_dob</Name>
        <Confidence>0.800000011920929</Confidence>
        <Value>04/26/1975</Value>
      </ExtractionField>
    </ExtractionPage>
  </ExtractionResult>
  <DocumentId>5c94122d456cfb0e1414ac58</DocumentId>
  <FaxMessageInboundSourceMetadata>
    <SourceId>ct18884445555-20190321153726799-281-10</SourceId>
    <UserAccountNumber>445489</UserAccountNumber>
    <MessageId>ct18884445555-20190321153726799-281-10</MessageId>
    <RecipientNumber>18884445555</RecipientNumber>
    <Pages>1</Pages>
    <Speed>99999</Speed>
    <MessageInitiated>2019-03-21T22:37:26Z</MessageInitiated>
    <RemoteCSID>Concord Fax</RemoteCSID>
    <Duration>3</Duration>
    <Status>Success</Status>
    <AccountNumber>445488</AccountNumber>
    <SenderNumber>13125554444</SenderNumber>
    <Size>70396</Size>
    <Resolution>2</Resolution>
    <Compression>0</Compression>
    <PartialFax>0</PartialFax>
    <FileFirstPage>0</FileFirstPage>
    <FilePageCount>0</FilePageCount>
    <IsStreaming>false</IsStreaming>
  </FaxMessageInboundSourceMetadata>
</Document>
```