



Net2Net REST Clv 6.0

CONTENTS

1. Net2Net REST	3
2. Integration & Production	5
3. Messages Overview	9
3.1. Clinical Direct Messaging Message Flow	10
3.2. Detailed Clinical Direct Messaging Workflows	11
4. Message Definitions	17
4.1. Net2Net Clinical Message	22
4.2. Processing Status Responses, Verifies, and Errors	26
5. Application Certification Requirements (For Message Validation Only)	30
6. Examples	32
7. Clinical Direct Messaging Processing Summary	40
8. Changes From Previous Versions	43
9. Disclaimers	44
10. Document Change Log	45

1. Net2Net REST

Product Overview

Net2Net REST is used to exchange Clinical messages. A Clinical message in the context of this guide is a message delivered to and from authorized customers on the Surescripts network (as such may be renamed from time to time by Surescripts), and satisfies the following requirements:

- Contains clinical or administrative information about a specific patient used for health care payment, treatment, or operations.
- Is not structured except to the extent necessary to transmit an attachment and as necessary for address fields.
- Is not limited to a specific message type or use case.
- Does not require message type or use case-specific workflow integration into the Electronic Health Record (EHR).

By implementing the Net2Net REST standard, Surescripts enables our customer to meet Meaningful Use Stage 2 (MU2) requirements. Surescripts messaging meets the transport and security specifications outlined in the Office of the National Coordinator's (ONC) Health Information Technology: Standards, Implementation, Specifications, and Certification Criteria for Electronic Health Record Technology, 2014 edition.

About This Guide

This Surescripts Net2Net REST Implementation Guide assists customers in developing and transferring messages needed to provide secure information exchange between providers. The audience for this document includes any customer responsible for developing a system interface to the Surescripts network using Net2Net REST.

This guide is meant to support, integrate, and further clarify the NCPDP standard schemas, implementation guides and best practices as used through the Surescripts network. It does not reproduce the base standard in its entirety.

Requirements beyond those in the NCPDP documentation are called out in this guide and are also enforceable. Customers should read and comprehend the associated standards prior to reading this guide.

Document References

Please reference the following documents when reading this Guide:

Surescripts-provided documents:

Surescripts_Clv6.0.xsd

Documents to be obtained by the customer:

NCPDP's SCRIPT Standard Implementation Guide (Version 10, Release 6)-pertinent to portions of the common message header, Status, Error, and Verify (<https://ncpdp.org/>)

NCPDP SCRIPT Implementation Recommendations Version 1.43 (<https://ncpdp.org/>)

NCPDP's External Code List 2017071 (NCPDP online tool) (<https://ncpdp.org/>)

2014 Edition Release 2 Electronic Health Record (EHR) Certification Criteria and the ONC HIT Certification Program; Regulatory Flexibilities, Improvements, and Enhanced Health Information Exchange; Final Rule.

This guide utilizes the NCPDP (National Council for Prescription Drug Programs) messaging standard Prescriber/Pharmacist Interface SCRIPT version 2023011 as a baseline. To become an NCPDP member, please go to

<https://www.ncdp.org/Membership.aspx>.

The NCPDP SCRIPT Standard 2023011 package contains a complete list of fields and additional requirements. Customers should read and comprehend the associated standards prior to reading this guide.

NCPDP is an American National Standards Institute (ANSI) accredited Standard Development Organization. The NCPDP "SCRIPT" standard is a copyrighted document and may be obtained by contacting:

NCPDP

9240 E. Raintree Drive

Scottsdale, AZ 85260-7518

Phone: (480) 477-1000

Fax: (480) 767-1042

<https://www.ncdp.org>

2. Integration & Production

The following section defines related technical and support elements needed to achieve certification for moving into production, along with our Integration, Compliance, and Certification processes.

Integration Process

Once Surescripts assigns an Integration resource, customers will be contacted to begin the project(s) and will be guided through the Surescripts Integration process. The Integration resource will provide Surescripts documentation, product education, access to the Surescripts staging environment, assistance during design and development milestones, and testing to prepare for message validation. The Integration resource will perform message validation testing and, once successful, will provide the customer access to the Surescripts production environment.

Note: The time frame of the project can vary depending on the customer's resource allocation for the project.

Meeting Requirements

During Integration, customers undergo message validation testing to demonstrate all requirements are met. Certification focuses on message format and, when appropriate, application workflow and display in accordance with Surescripts documentation and the associated Application Certification Requirements (ACRs). Upon successful completion of message validation and, when applicable, other pre-production network requirements (e.g., Identity Proofing, DEA audit), customers will be enabled in production.

For requirements, consider the following:

- Surescripts ACRs are required to be met to achieve production status on the Surescripts network and will be enforced as part of certification.
- Surescripts business rules are data validations applied by Surescripts beyond the schema requirements that will cause a message to be successful or rejected.
- Surescripts test cases do not cover all possible scenarios in production. Customers are responsible for testing all scenarios specific to their production environment.
- In accordance with the customer's legal agreement with Surescripts, each customer is responsible for ensuring compliance with all applicable laws including, but not limited to, local and state laws/regulations where doing business.

Transition to Production

Once message validation and the contract are complete and approved by the Surescripts Certification Review Board, the customer is ready to move into production. Surescripts will configure the production connection and validate successful operations with the customer. Prior to the transition to production, Surescripts Account Management will work with the customer and internal Surescripts teams to discuss the following:

- Production support contacts (Escalation Matrix)
- Support process and training
- Support hours

Note: The transition to production is concluded with a Lessons Learned discussion and/or Satisfaction Survey.

Terminology Usage

For terminology usage throughout this Guide, consider the following:

Term	Term usage
must	Requirements that are enforced as part of the production code or Surescripts business rules.
shall	The requirements customers are required to meet in order to be certified on the Surescripts network. These requirements will be enforced as part of certification.
should	Used for guidance and best practices. Best practices can also be found in Best Practice sections. Customers are encouraged, but not required, to meet best practices in order to be certified on the Surescripts network.
C.##	This designates a PMBC workflow application certification requirement.

Communication Rules

Please refer to the Connectivity and Authentication Guide for additional connectivity and authentication information. For the network to be reliable, there are communication rules to which all customers must adhere.

Timeouts

For timeouts, consider the following:

- When sending a message to Surescripts, the initiator should set the http timeout to 30 seconds.
- The Receiving System must reply with a valid Status/Error response within 24 seconds.
- When a Status 000 is sent as a reply to a Clinical Message, the Receiving System must follow-up with a Verify or Error message within 55 minutes, because Surescripts may return a timeout error if no Verify or Error message has been received.

UTC Time Format

By using Coordinated Universal Time (UTC), the receiver of a message will know the time regardless of their time zone. For example: If a message was sent from Boston at 5:30PM EDT (Eastern Daylight Time), the time would be sent as 21:30 UTC time. If this message was received in Chicago CDT (Central Daylight Time), the 21:30 UTC could be converted to the local CDT time of 4:30PM. Refer to http://en.wikipedia.org/wiki/Coordinated_Universal_Time, or <http://www.w3.org/XML/> for more information. Synchronize UTC time with NIST (National Institute of Standards and Technology). Drift of no more than one minute will be acceptable.

When sending only a Date (not Date and Time), it should be sent in your local time zone, and the receiver should interpret it in their local time. Neither party should attempt to convert the Date to UTC.

All standard programming languages should have a function for generating a date in the UTC time zone or displaying a date in the local time zone. The format of the date/time fields in the XML schema must use the xsd:dateTime format. Examples of that format are: CCYY-MM-DDTHH:MM:SS.FZ, where the UTC time zone may be specified as Z, + 00:00, or - 00:00. For example, 2013-01-01T16:09:04.5Z, or 2013-01-01T16:09:04.5-00:00, or 2013-01-01T16:09:04.5+00:00, where 16:09:04.5Z would be 16 hours, 09 minutes, 04 seconds, 5 fractional seconds. UTC time is denoted by either the "Z" in the first example or the "-00:00" in the second example, or the "+00:00" in the third example. The fractional seconds is not required. Refer to xsd:dateTime for more information.

For simple date fields that do not include the time portion, the format is: CCYY-MM-DD.

Character Set

The character set contains ASCII values 32 – 126, which include:

Symbols	! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ ` { } ~
Numerals	0 to 9
Letters, upper and lower case	A to Z, a to z

Unprintable characters, such as control characters, are not used within the field sets. Defined unprintable characters are used as delimiters.

UTF-8 is the required character encoding for XML. Other encoding formats are not supported by Surescripts. It is recommended that customers declare their UTF-8 formatting in the message header.

Handling Optional Fields

Customer shall be able to process any valid message request or response that they are certified on (conditional/optional, valid data elements and values shall not cause processing failure of the message).

Numeric Representation

The decimal point is represented by a period and shall be used as follows:

- Only when there are significant digits to the right of the decimal.
- When there is a digit before and after the decimal point.
- Not with whole numbers.

For example, consider the following possible values:

Correct	2.515	251.5	25.15	2515	0.2515	2.5
Incorrect	.2515	2515.	3.00			

Requirement Designation

Segment Attributes:

Code	Description
M	Required/Mandatory - the segment must be used.
C	Situational/Conditional - the segment must be used if conditions are met. Some fields may not have specific conditions. Data should be sent if available. Where comments are "Not used by Surescripts", information will be passed on, but not used, by Surescripts for processing.
NU	Not used or for future use.

Note: Elements that are grouped together in a composite may be marked as mandatory; however, if the group itself is marked as conditional, then these are only required if you use the group.

In the example below, the DocumentType and DocumentData are mandatory only if the File is sent.

Element Name	Code	Comments
File	conditional	This Type of attachment can be used by the send to include a file attachment.
DocumentType	mandatory	If a file attachment is included, then a DocumentType must be specified and should be a valid MIME type. For example: application/pdf
DocumentVersion	conditional	Optional version number for the document type
DocumentData	mandatory	The Base64 Encoded string of the file attachment

Compliance

Surescripts goal is efficiency and consistency across the network so that all customers can meet the highest measures of patient safety, end-to-end reliability, and quality. To ensure that customers comply with, and adhere to, the approved certification requirements, Surescripts:

- Initiates a remediation process for identified compliance issues.
- Monitors customers in production to ensure all network customers remain in compliance with certification requirements and contractual terms.

Customers agree to notify Surescripts when they have altered, reconfigured, or disabled any portion of a Surescripts certified software product or module, before moving such changes into production, as they may create a circumstance of non-compliance with the Surescripts certification issued. In those instances, Surescripts will work with the customer to perform a timely re-certification, if required, to ensure network compliance and safety.

This guide is intended for certification on our network only and is not intended to ensure compliance with state and federal law. In accordance with the customer's legal agreement with Surescripts, each customer is responsible for conducting its own due diligence to ensure compliance with all applicable laws and requirements, including, but not limited to, local and state laws and regulations in which the customer's application is deployed and used.

As a reminder, Surescripts conducts certification with customers to ensure the application adheres to network requirements. Surescripts will enforce mandatory fields as required by the Standards body and Surescripts guide requirements. To maximize interoperability, customers are recommended to support optional fields that have been created to address gaps in discrete data needs and the many solutions that are in place for the benefit of the receiver. Surescripts encourages, but does not guarantee, the use of optional discrete fields to support end user workflows.

3. Messages Overview

Message and Response Descriptions

Clinical Message

A *Clinical* message is used to communicate patient-related clinical information between health care providers. It may contain textual data as well as, structured content, and file attachments.

Status

This response indicates the *Clinical* message was accepted for processing by receiving system. Receiving system automatically generates the *Status* response to the *Clinical* message originator. The *Clinical* message originator is not required to perform any additional action upon receipt of the *Status* response.

- *Status 010* indicates the message was delivered electronically.
- *Status 000* indicates that the message was accepted for processing, but final status is not yet available.

Verify

This message indicates a *Clinical* message was delivered electronically to the recipient. The receiving system of the *Verify* message (*Clinical* message originator) is responsible for providing a *Status* response to the *Verify* message originator.

Error

This message/response indicates that an error has occurred and the message has not reached its intended recipient.

- An *Error* response is automatically generated when there is a communication problem delivering the *Clinical* message to the receiving system and the request was terminated.
- An *Error* message is generated when the receiving system is unable to deliver the *Clinical* message to the intended recipient.

3.1. Clinical Direct Messaging Message Flow

The Surescripts Clinical Direct Messaging service enables secure communication of clinical information between providers of healthcare services utilizing the Surescripts network. Network-based routing and electronic message delivery is included with this service. Customers may extend their delivery options by contracting for Guaranteed Delivery of *Clinical Direct* messages. Guaranteed Delivery adds fax and postal mail delivery of *Clinical Direct* messages to the recipient of the electronic message.

Electronic Routing

A *Clinical* message can be delivered to specified recipients electronically. Electronic message routing uses HTTP (Hypertext Transfer Protocol), a request-response protocol. For each message that is sent, a response message is returned. The status of the delivery of the message is defined by the type of the response (e.g., 000 = pending, 010 = success, 600 = error, etc.).

Within the HTTP request-response protocol, it is preferred that the response from the request be a *Status 010* or *Error* response instead of a delivery of the *Status 000* response with a subsequent freestanding Verify/Error.

The maximum allowable size for a message, including attachments, is restricted to 20MB.

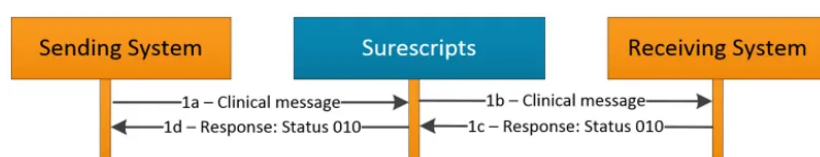
3.2. Detailed Clinical Direct Messaging Workflows

Glossary of workflow terms:

- **Sending System:** System connected to Surescripts network originating *Clinical* message.
- **Sender:** Originator of *Clinical* message routed through sending system.
- **Receiving System:** System connected to Surescripts network receiving original *Clinical* message.
- **Recipient:** Receiver of the original *Clinical* message within the receiving system.

Single Recipient

Receiving System's Response Status 010



Scenario 1:

Sender generates a *Clinical* message.

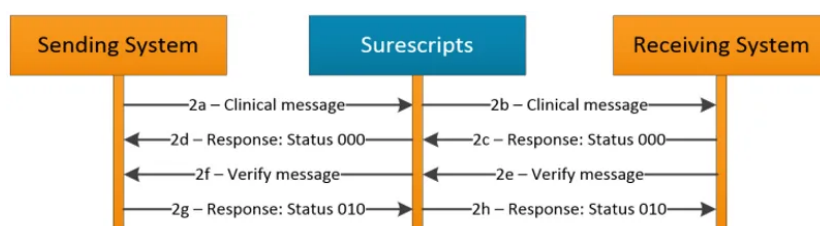
1a). Sending System sends *Clinical* message to Surescripts.

1b). Receiving System receives the *Clinical* message forwarded by Surescripts.

1c). Receiving System sends a *Status 010* response to Surescripts to indicate that the *Clinical* message has been delivered.

1d). Sending System receives the *Status 010* response forwarded by Surescripts.

Receiving System's Verify Message



Scenario 2:

Sender generates a *Clinical* message.

2a). Sending System sends a *Clinical* message to Surescripts.

2b). Receiving System receives the *Clinical* message forwarded by Surescripts.

2c). Receiving System responds with a *Status 000* response to Surescripts indicating the *Clinical* message was accepted for processing.

2d). Sending System receives the *Status 000* response forwarded by Surescripts.

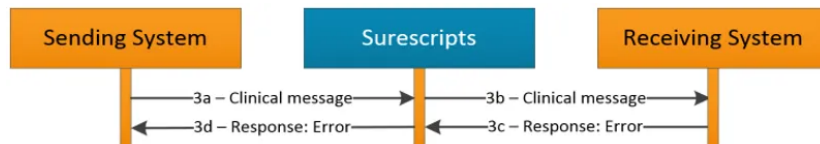
2e). Receiving System sends a *Verify* message to Surescripts indicating the *Clinical* message was successfully delivered electronically to Recipient.

2f). Sending System receives the *Verify* message forwarded by Surescripts.

2g). Sending System responds to the *Verify* message with a *Status 010* response to Surescripts.

2h). Receiving System receives the *Status 010* response forwarded by Surescripts.

Receiving System's Error Response



Scenario 3:

Sender generates a *Clinical* message.

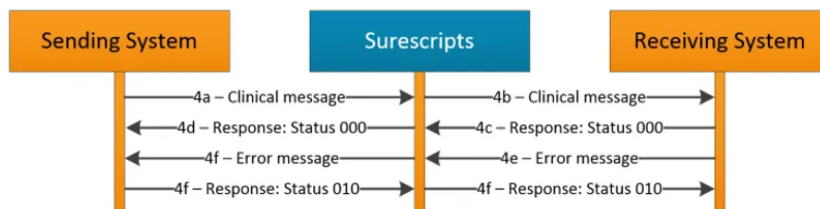
3a). Sending System sends *Clinical* message to Surescripts.

3b). Receiving System receives the *Clinical* message forwarded by Surescripts.

3c). Receiving System responds with an *Error* response to Surescripts.

3d). Sending System receives the *Error* response forwarded by Surescripts.

Error Message



Scenario 4:

Sender generates a *Clinical* message.

4a). Sending System sends a *Clinical* message to Surescripts.

4b). Receiving System receives the *Clinical* message forwarded by Surescripts.

4c). Receiving System sends a *Status 000* response to Surescripts.

4d). Sending System receives the *Status 000* response forwarded by Surescripts.

4e). Receiving System sends an *Error* message to Surescripts.

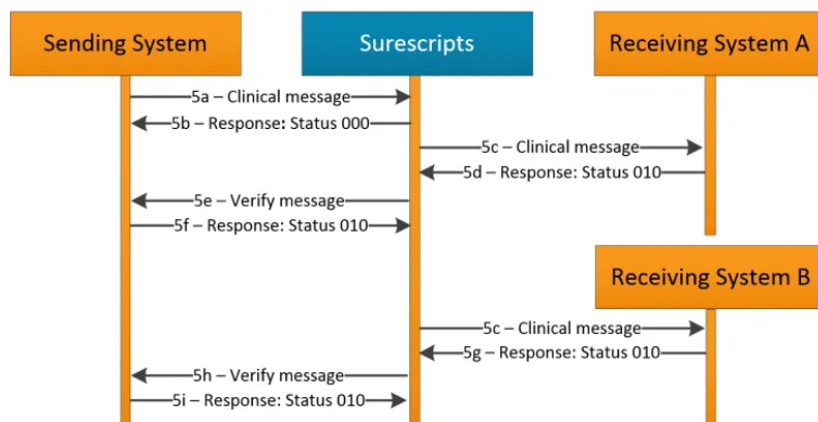
4f). Sending System receives the *Error* message forwarded by Surescripts.

4g). Sending System responds to the *Error* message with a *Status 010* response to Surescripts.

4h). Receiving System receives the *Status 010* response forwarded by Surescripts.

Multiple Recipients

Receiving Systems' Status 010 Response



Scenario 5:

Sender generates a *Clinical* message.

5a). Sending System sends a *Clinical* message to Surescripts.

5b). Sending System receives a *Status 000* response generated by Surescripts for all multiple recipient messages.

5c). Receiving Systems A and B receive the *Clinical* message forwarded by Surescripts.

5d). Receiving System A responds with a *Status 010* response to Surescripts.

5e). Sending System receives a *Verify* message generated by Surescripts.

5f). Sending System responds with a *Status 010* response to Surescripts. (Surescripts does not forward response to Receiving System.)

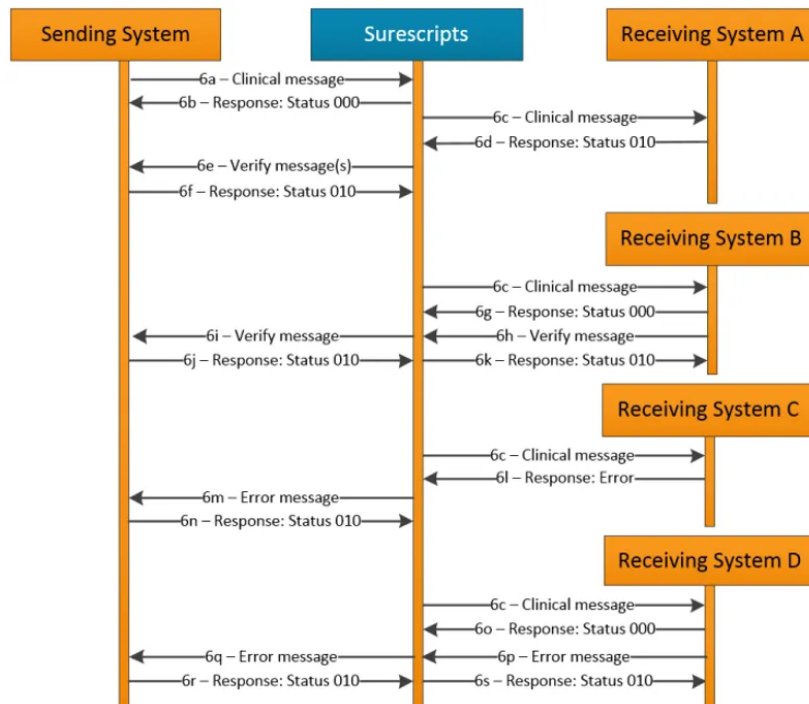
5g). Receiving System B responds with a *Status 010* response to Surescripts.

5h). Sending System receives a *Verify* message generated by Surescripts.

5i). Sending System responds to the *Verify* message with a *Status 010* response to Surescripts. (Surescripts does not forward response to Receiving System.)

Note: Receiving systems' responses can occur in any order.

Multiple Recipient Message with Errors and Verifies



Scenario 6:

Sender generates a *Clinical* message.

6a). Sending System sends a *Clinical* message to Surescripts.

6b). Sending System receives a *Status 000* response generated by Surescripts for all multiple recipient messages.

6c). Receiving Systems A, B, C, and D receive the *Clinical* message forwarded by Surescripts.

6d). Receiving System A responds with a *Status 010* response to Surescripts.

6e). Sending System receives a *Verify* message generated by Surescripts.

6f). Sending System responds with a *Status 010* response to Surescripts. (Surescripts does not forward response to Receiving System.)

6g). Receiving System B responds with a *Status 000* response to Surescripts indicating the *Clinical* message was accepted for processing (not forwarded to Sending System).

6h). Receiving System B sends a *Verify* message to Surescripts.

6i). Sending System receives the *Verify* message forwarded by Surescripts.

6j). Sending System responds to the *Verify* message with a *Status 010* response to Surescripts.

6k). Receiving System B receives *Status 010* response forwarded by Surescripts.

6l). Receiving System C sends an *Error* response to Surescripts.

6m). Sending System receives an *Error* message generated by Surescripts.

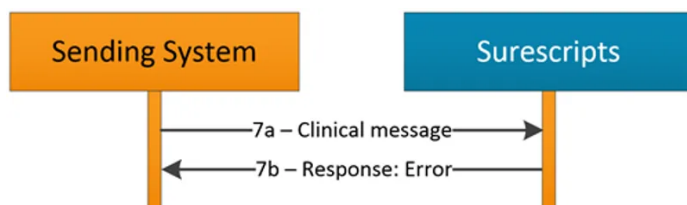
6n). Sending System responds to the *Error* message with a *Status 010* response to Surescripts (not forwarded to Receiving System C).

- 6o). Receiving System D responds with a *Status 000* to Surescripts.
- 6p). Receiving System D sends an *Error* message to Surescripts.
- 6q). Sending System receives the *Error* message forwarded by Surescripts.
- 6r). Sending System responds to the *Error* message with a *Status 010* response to Surescripts.
- 6s). Receiving System D receives the *Status 010* response forwarded by Surescripts.

Note: Receiving Systems' responses can occur in any order.

Undeliverable Messages

Surescripts' Error Response



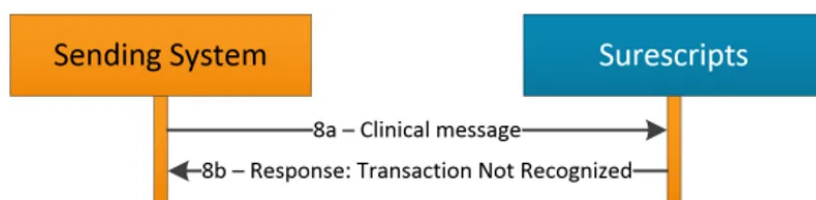
Scenario 7

Sender generates a *Clinical* message.

- 7a). Sending System sends *Clinical* message to Surescripts.
- 7b). Sending System receives an *Error* response from Surescripts.

Note: This scenario applies to both single and multiple recipient messages.

Surescripts: Transaction Not Recognized



Scenario 8:

Sender generates a *Clinical* message.

- 8a). Sending System sends a *Clinical* message to Surescripts.
- 8b). Sending System receives *Transaction Not Recognized* response from Surescripts.

Message Validation

Surescripts will ensure that customers are in compliance with the message specifications outlined in this guide during testing and will continue to enforce once in production.

At a minimum, Surescripts validations include:

- XML schema validation
- The sender identification and authentication
- The recipient identification
- Syntax of the message, including field lengths, data types, and code values
- Surescripts business rules

Note: Surescripts ACRs are not enforced as part of validations, but instead through the certification process.

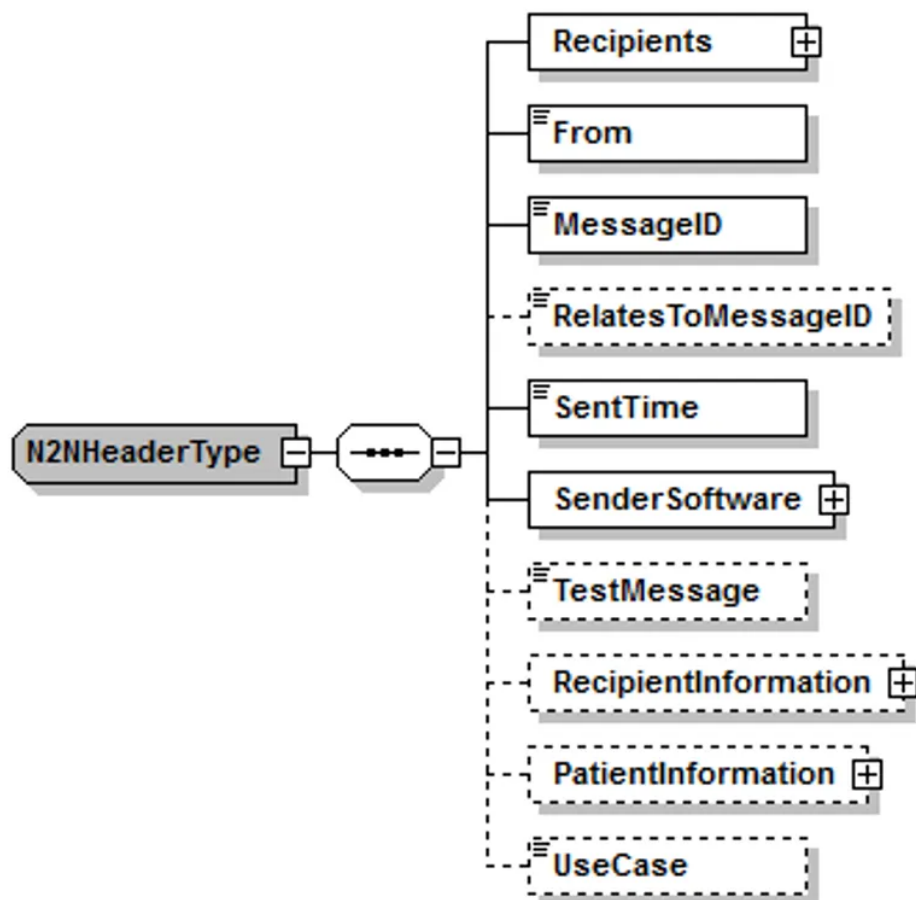
4. Message Definitions

Message Header

Surescripts uses a message header for all messages. The message header contains information required for routing of the message and is separate from the clinical content of the message body. The XML schemas are available as a supplemental file.

Message Header – XML

In XML diagrams, blocks that are solid are mandatory, whereas blocks that have dotted lines are conditional.



Note: RecipientInformation is used only for guaranteed delivery.

Patient Information

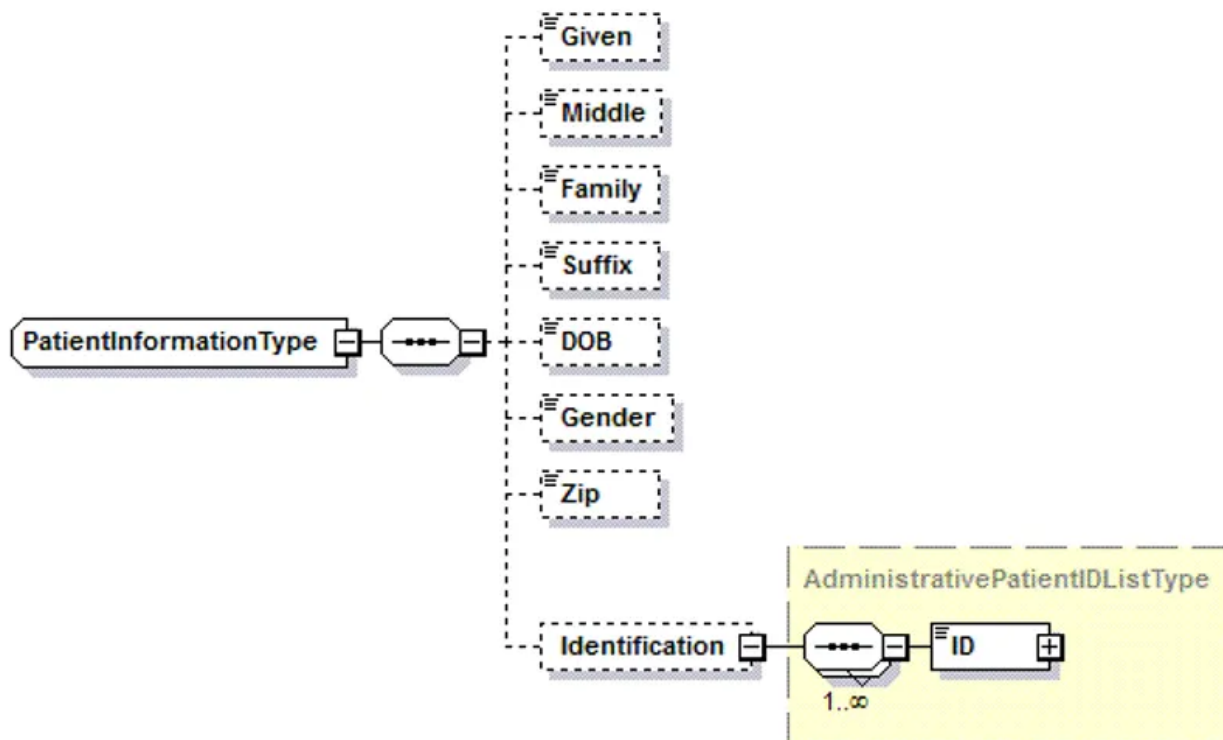
Clinical messages often relate to the care of a particular patient. Customers should include patient identification data with the message to assist the recipient's software in matching the patient to a local patient record.

When sending a new message, a customer should include <PatientInformation> in the message, which should include, at minimum, the <Given>, <Family> elements and should include the <DOB>, <Gender>, and <Zip> elements, as well as a valid occurrence of the <Identification> element.

Note: Surescripts does not guarantee the patient identifier provided in the message is unique or specific to the recipient's organization. When receiving messages, the patient identifier should not be used exclusively to match the clinical information provided in the message with a patient within the recipient's organization.

A message should contain data for a single patient. If a message is not related to a single patient the sender should not specify any <PatientInformation> in the message.

Patient Information



Message Header Elements

M/C Mandatory or Conditional

Element	Cod e	Comments
N2NMessage/@version	mandatory	Message version number. Value = Clv6
N2NMessage/@release	mandatory	Message release number. Value = 0
Recipients	mandatory	List of all receivers of the message.
To	mandatory	The identification of the intended receiver of the message.
To/@Qualifier	mandatory	Value = DIRECT
To/@CC	conditional	Indicates that the recipient is a secondary (informational) recipient of the message. CC = true If not true, do not include this attribute.
To/@FriendlyName	conditional	Value identifies a user friendly name of the recipient used for display.
To/@Route	conditional	Only added by Surescripts to -indicate that the receiving system is responsible for responding for the identified recipient. Route = true If not true, Receiving System is not responsible for responding to the message.
From	mandatory	The identification of the sender of the message. See Multiple Recipient Messages for more information.
From/@Qualifier	mandatory	Valid Values: DIRECT ZZZ - only used when synchronously responding to a multiple recipient message
From/@FriendlyName	conditional	Value identifies a user friendly name of the sender used for display.
MessageID	mandatory	The sender should ensure the combination of the From and MessageID elements is unique for at least 18 months. Surescripts recommends an unformatted Globally Unique Identifier (GUID).
RelatesToMessageID	conditional	On a responding message, this field is mandatory for the Error, Status, and Verify messages, and is used to link the original message to the response.

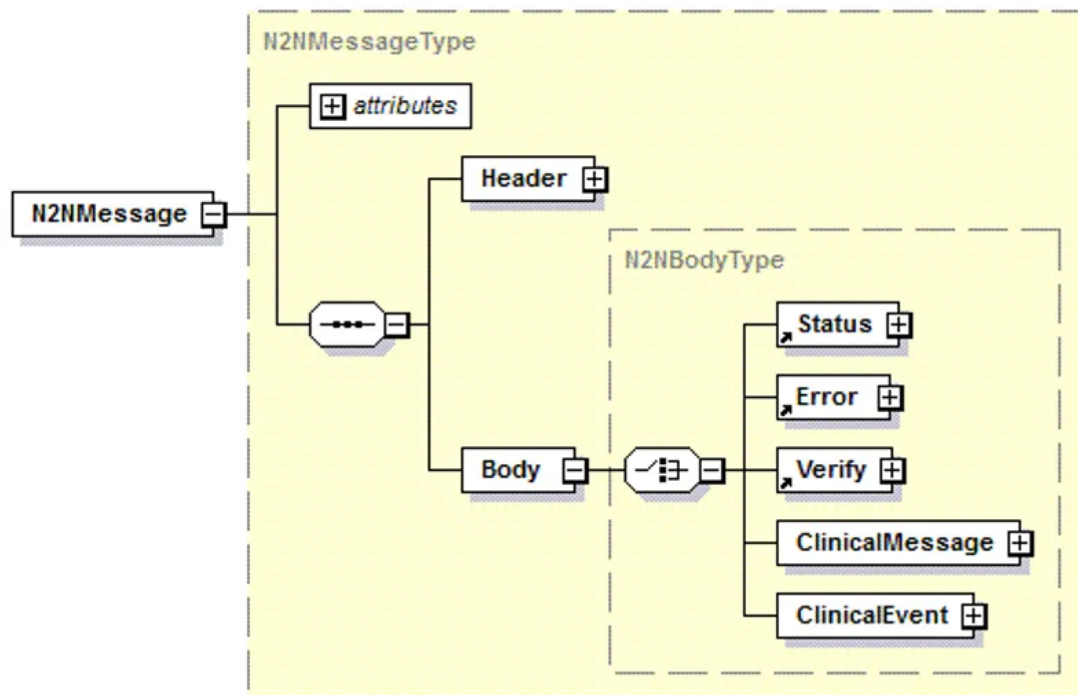
Element	Cod e	Comments
	al	
SentTime	man dato ry	Date of the transmission. Please see UTC Time Format section.
SenderSoft ware	man dato ry	Identifies the entity responsible that generated the message. Contains the software vendor name, product, and version/release.
SenderSoft wareDevelo per	man dato ry	Identifies the entity responsible for the software that generated the message. The developer may be a software vendor or, if the software was developed "in-house", the developer is the entity actually sending the message (e.g., a chain). The value transmitted is determined by the Sender Software Developer.
SenderSoft wareProduc t	man dato ry	The text field that identifies the software developer's product, as determined by the software developer.
SenderSoft wareVersion Release	man dato ry	The text field that identifies the version and release of the software product, as determined by the software developer.
TestMessag e	cond ition al	Do not include this element in production messages.
RecipientInf ormation	man dato ry	Used for Guaranteed Delivery only.
PatientInfor mation	cond ition al	Contains demographic information on the patient that is associated with the clinical message.
Given	cond ition al	Patient's given (first name)
Middle	cond ition al	Patient's middle names(s)
Family	cond ition al	Patient's family (last name)
Suffix	cond ition al	Patient's name suffix

Element	Code	Comments
DOB	conditional	Patient's Date of Birth in format YYYY-MM-DD where YYYY indicates the year: MM indicates the month and DD indicates the day.
Gender	conditional	Patient's Administrative gender code. Valid Values: F = Female M = Male O = Other U = Unknown
Zip	conditional	Patient's zip code in format of ##### or ##### (5 or 9 digit). Please be aware that no hyphen is to be included.
Identification	conditional	A collection of one or more patient identifiers. For each occurrence, both <ID> and @Issuer are required.
ID	mandatory	A patient identification string.
ID/@Issuer	mandatory	A globally unique identifier representing the issuing authority for the Patient ID.
UseCase	conditional	Do not send for Clinical Direct Messaging. Supplemental guides may contain additional instructions for this field. Customers may receive messages with this field populated.

4.1. Net2Net Clinical Message

A *Clinical* message is used to communicate patient related clinical information between health care providers. It may contain textual data, as well as structured content and file attachments.

Net2Net Message XML Diagram



Sending a Net2Net Clinical Message

To send a Net2Net *Clinical* message, the customer must

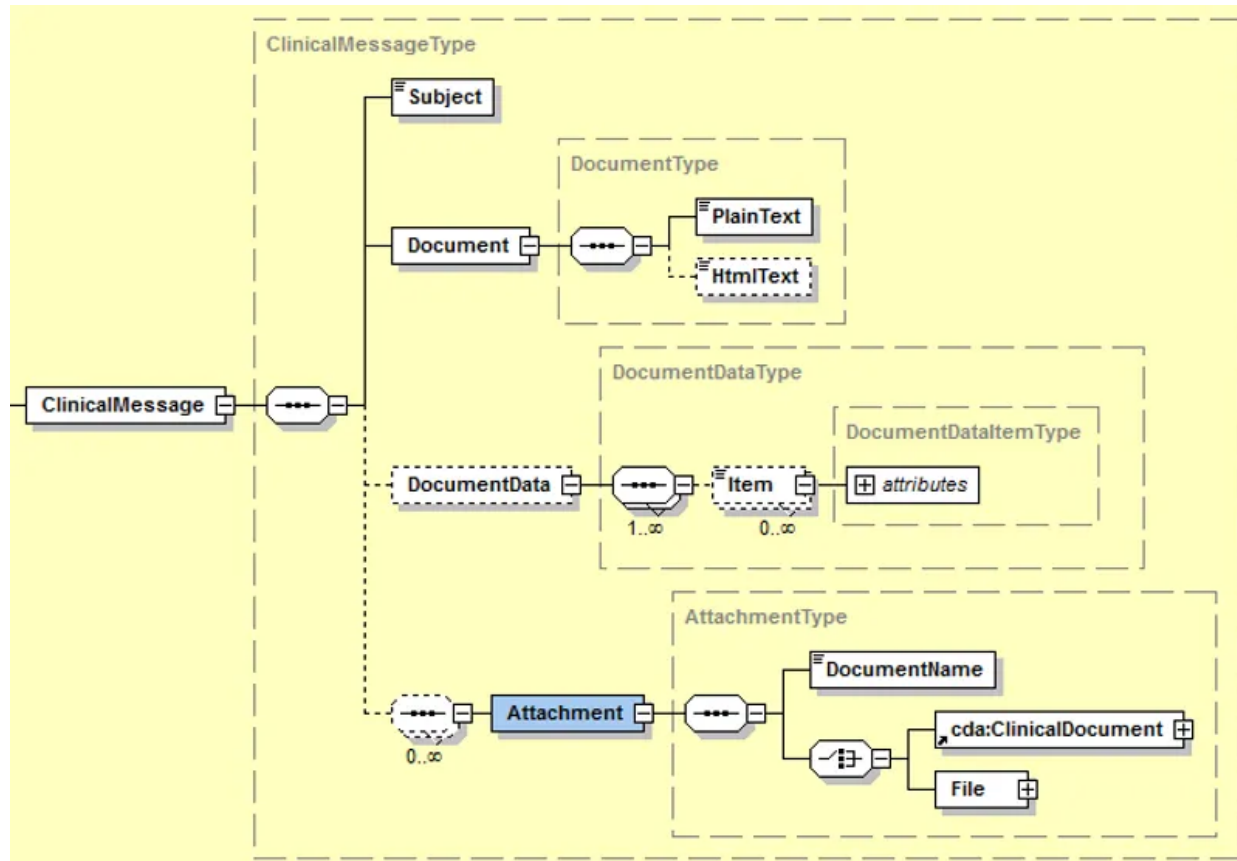
1. Collect message content, attachments, and the recipient information;
2. Validate that the message conforms to the schema; and
3. Call the Surescripts API to submit the *Clinical* message(s).

Once a *Clinical* message is sent, the Surescripts API provides a synchronous response to the submission of a message. This response is a Status message "010" if the message was received and delivered; a Status message "000" if the message was accepted for processing, or an Error message if the message could not be accepted for processing.

Receiving a Clinical Message

To receive a message, customers must set up a server to which Surescripts can send messages. Surescripts will send all messages to the customer as an HTTP POST. The customer system must provide a synchronous response to messages and freestanding Error/Verify messages. See [Detailed Clinical Messaging Workflows](#) for more information.

Clinical Document Message XML Diagram



Net2Net Clinical Message Body Elements

M/C Mandatory or Conditional

Element	Code	Comments
ClinicalMessage	mandatory	
Subject	mandatory	The subject of the <i>Clinical</i> message. Protected Health Information (PHI) under HIPAA must not be included in the subject.
Document	mandatory	Required if sending a <i>Clinical</i> message in the message body.
PlainText	mandatory	Example: "ABC Clinic has provided services to the patient named in the attached document. The patient has identified you as their primary care physician."
HtmlText	conditional	This is optional and allows for content formatting.
DocumentData	conditional	Specifies the additional data in the message. This consists of a collection of items, with a key attribute and the value. Ignore values not supported by your system. For example: <pre><DocumentData> <Item key="x-kryptiq-notification">onetime</Item> <Item key="x-kryptiq-message-class-type">Standard</Item> <Item key="x-kryptiq-trackingid">6dabf2a36fa4406eacd09ce23190ceb1</Item> <Item key="x-kryptiq-remotemessagekey">jzcs3dnchvrhbhdvso2eatloj4molg</Item> </DocumentData></pre> Note: Empty tags are permitted for use in the DocumentData element.
Item	mandatory	Specified data value.
Item/@key	mandatory	Identifier of the specified data value.
Attachment	mandatory	A collection of one or more attachments. Each occurrence consists of a DocumentName and one (only one) of the document elements.
DocumentName	mandatory	Filename of the document. The file extension should be included. Example: visitsummary.pdf
cda:ClinicalDocument	conditional	The sender can use this type of attachment to include a CDA attachment. Note: Inline XML is the preferred method for passing the CDA.
File	conditional	The sender can use this type of attachment to include a file attachment.
DocumentType	mandatory	If a file attachment is included, then a <DocumentType> must be specified and should be a valid MIME type. For example: application/pdf

Element	Code	Comments
DocumentVersion	conditional	Optional version number for the document.
DocumentData	mandatory	The Base64 Encoded string of the file attachment.

4.2. Processing Status Responses, Verifies, and Errors

Status responses, Verifies, and Errors communicate current information regarding the delivery of a *Clinical* message.

The Receiving System should respond to a message for a single recipient as follows:

- If the message was received and delivered to the recipient, the customer should respond with *Status 010*, which indicates that all processing is complete. Customers may not send subsequent *Error* or *Verify* messages relating to the clinical message.
- If the message was received and time is needed to process, the customer should respond with *Status 000*. This must be followed within 55 minutes with either:
 - a *Verify* message or
 - an *Error* message with the appropriate Error Code.

The Sending System should respond to a *Verify* message or *Error* message as follows:

- If the *Verify/Error* message was received successfully, the Sending System should respond with a *Status 010*.
- If an error occurred when receiving the *Verify/Error* message, the Sending System should respond with an *Error* message and the appropriate Error Code.

Multiple Recipient Messages

The Receiving System responds to multiple recipient messages where each recipient's *Route* = true with one of the following HTTP responses:

- If the message was received and delivered to *all Route* = true recipients, the Receiving System should return a single *Status 010* response indicating the processing is complete for those recipients. The Receiving System should not send subsequent *Verify* or *Error* message(s).
- If the message was received and time is needed to process at least one recipient by the Receiving System, the Receiving System should respond with a *Status 000* response. This *Status 000* response must be followed with a *Verify* or *Error* message for each recipient where the *Route* = true detailing the final status of the message.
- If the same error occurred for *all Route* = true recipients, the Receiving System should return an *Error* response and the appropriate Error code for those recipients. The Receiving System should not send subsequent *Verify* or *Error* message(s).

When responding to a message addressed to multiple recipients, hosted by the Receiving System, the Receiving System should use the "From" field Qualifier of "ZZZ" with a field Value of the Receiving System's Portal Name (i.e., <From Qualifier="ZZZ">PortalName</From>).

Note: On multiple recipient messages, Surescripts responds with a *Status 000* response and <From Qualifier="ZZZ">Surescripts</From>

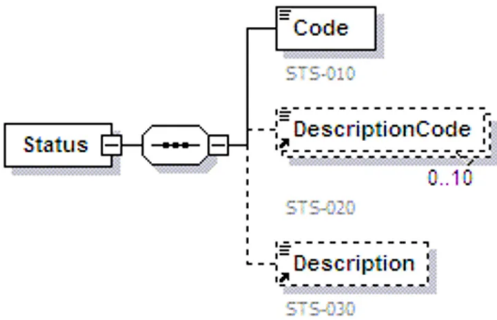
Status

This response indicates the *Clinical* message was accepted for processing or the *Clinical* message was successfully delivered.

The HTTP *Status 010* response indicates the process is complete. Subsequent acknowledgement messages should not be sent.

The HTTP *Status 000* response indicates further processing is required. Subsequent acknowledgement messages are required.

Status XML Diagram



Status Elements

M/C Mandatory, Conditional, or Not used

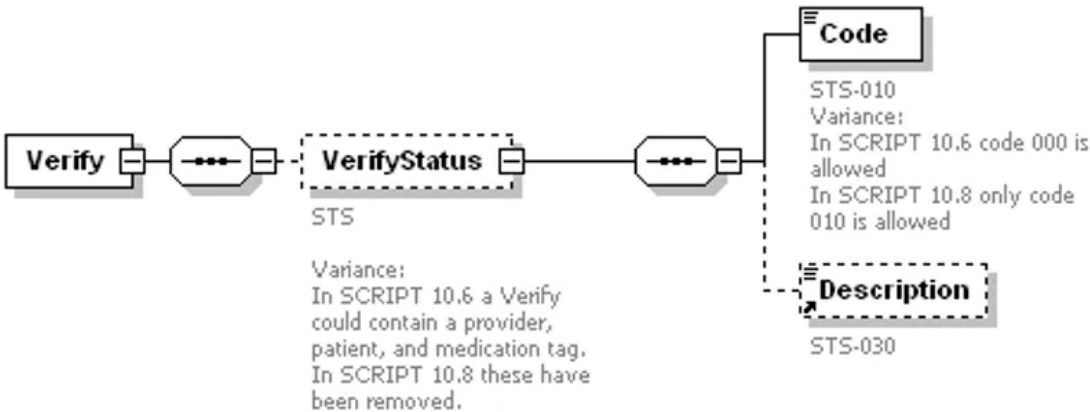
Element	Code	Comments
Status	mandatory	Message type.
Code	mandatory	Status Codes used to relay successful communications. 000 = Transaction successful; pending 010 = Successfully accepted by ultimate receiver
DescriptionCode	conditional	Repeats up to 10 times. C = Status (this field is conditional for Status) Reject Codes used by responder who takes responsibility for the message. A complete list can be found in the NCPDP External Code List (X12 DE 1131 Reject Code STS Segment).

Verify

This message indicates the receiving system has successfully delivered the message to the recipient's mail box. The *Verify* message is not to be used to indicate that the *Clinical* message has been opened/read.

A Verify message indicates the process is complete.

Verify XML Diagram



Verify Elements

M/C Mandatory, Conditional or Not used

Element	Code	Comments
Verify	mandatory	Message type.
Code	mandatory	Verify Code used to relay successful communications. 010 = Successfully accepted by ultimate receiver
DescriptionCode	conditional	Repeats up to 10 times. NU = Verify (this field is not used for Verify) Reject Codes used by responder who takes responsibility for the message. A complete list can be found in the NCPDP External Code List (X12 DE 1131 Reject Code STS Segment).

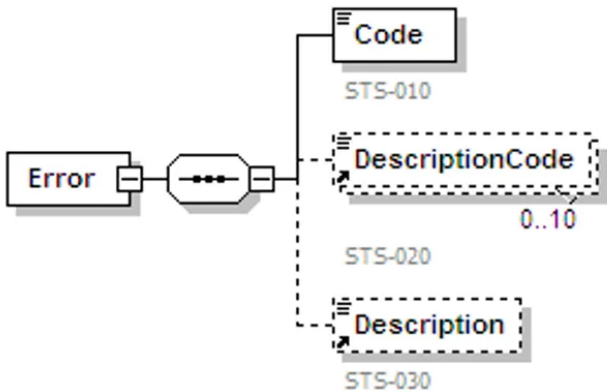
Error

This message/response indicates that an error has occurred and the message has not reached its intended recipient. An error can be generated when there is a communication problem or when an error occurs during the processing of the message. An *Error* can either be free-standing message or an HTTP response.

An *Error* response indicates the process is complete. Subsequent acknowledgement messages are not expected.

An *Error* message indicates the receiving system is unable to deliver the *Clinical* message to the intended recipient.

Error XML Diagram



Error Elements

M/C Mandatory, Conditional or Not used

Element	Code	Comments
Error	mandatory	Message type.
Code	mandatory	Error Codes used to relay rejected communications. 600 = Communication problem - try again later 601 = Receiver unable to process – do not retry 602 = Receiver System Error – try again later 900 = Transaction rejected – do not retry
DescriptionCode	conditional	Repeats up to 10 times. C = Error (this field is conditional for Error) Reject Codes used by responder who takes responsibility for the transaction. A complete list can be found in the NCPDP External Code List (X12 DE 1131 Reject Code STS Segment).
Description	conditional	Description of Error(s)

5. Application Certification Requirements (For Message Validation Only)

Note: The requirements listed here only apply to Clinical message validation.

Directory Presentation

C.100 The participant's application shall display, at a minimum, the following Provider data elements from the directory whenever browsing the directory or displaying directory search results.

- LastName
- FirstName
- ClinicName
- SpecialtyType1

C.101 The participant's application shall be able to display, at a minimum, the following Provider data elements from the directory.

- PrefixName
- LastName
- FirstName
- MiddleName
- SuffixName
- ClinicName
- AddressLine1
- AddressLine2
- City
- State
- Zip
- PhonePrimary
- SpecialtyType1

Sending Messages

C.102 The participant's application shall actively notify the end user when a message is not successfully sent to the recipient(s).

Examples of actively notifying the end user – A new message delivered to the end user, a flag on the message the notification relates to, a pop up box.

C.103 If the participant's application allows the end user to set or modify the message's subject line, the application shall display, without end user action, a disclaimer that Protected Health Information (PHI) under HIPAA must not be included in a message's subject line.

Sample disclaimer – Note: subject text is not encrypted - do not include sensitive or private information.

C.104 If the participant systematically populates the message's subject line, Protected Health Information (PHI) under HIPAA must not be included in a message's subject line.

C.105 If batching messages, the participant's application shall wait a maximum of one minute between processes to send messages to the Surescripts network.

Receiving Messages

C.106 The participant's application shall make the message available to the intended recipient(s) within one minute of receiving it.

C.107 The participant's application shall be able to receive a message of at least 20 MB.

C.108 The participant's application shall be able to display, at a minimum, the following from the message.

- Date/Time
- Sender (From)
- Subject
- Message Body (text/plain or text/html part)

C.109 The participant's application shall be able to display, at a minimum, the following attachment types:

- HTML
- RTF
- PDF
- TXT
- XML

C.110 The participant's application shall, at a minimum, make all attachments regardless of type available to download unless the attachment type is filtered for security reasons.

C.111 If an attachment is filtered for security reasons, the participant's application shall notify the recipient of the message the attachment was removed and the filename of the attachment.

6. Examples

Examples provided throughout this guide are not intended to be all-inclusive. This pertains to example workflows, element-specific (field) examples, or message examples. Customers should not restrict coding to the examples used herein.

When developing, this guide should be used along with the accompanying schema file, additional documentation referenced, and applicable customary XML coding standards.

Clinical Message with PDF Attachment

The following is an example of a *Clinical* message with a PDF attachment.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<N2NMessage version="Civ6" release="0" xmlns="http://www.surescripts.com/messaging">
  <Header>
    <Recipients>
      <To Qualifier="DIRECT" FriendlyName="User Two">usertwo@exampledomain.com</To>
      <To Qualifier="DIRECT" CC="true" FriendlyName="User Three">userthree@exampledomain.com</To>
    </Recipients>
    <From Qualifier="DIRECT" FriendlyName="User One">userone@exampledomain.com</From>
    <MessageID>f9733ed92efe46658389d1fab1fb3e50</MessageID>
    <SentTime>2018-11-13T15:27:41.184Z</SentTime>
    <SenderSoftware>
      <SenderSoftwareDeveloper>test developer</SenderSoftwareDeveloper>
      <SenderSoftwareProduct>test product</SenderSoftwareProduct>
      <SenderSoftwareVersionRelease>1.0</SenderSoftwareVersionRelease>
    </SenderSoftware>
    <PatientInformation>
      <Given>FirstName</Given>
      <Family>LastName</Family>
      <DOB>1960-06-29</DOB>
      <Gender>F</Gender>
      <Zip>55402</Zip>
      <Identification>
        <ID Issuer="1.3.6.1.4.1.34488.150.21">1054345A</ID>
        <ID Issuer="72BD5F4A-0F71-425E-B898-B18EE433A2FF">MR-1332-0001</ID>
      </Identification>
    </PatientInformation>
  </Header>
  <Body>
    <ClinicalMessage>
      <Subject>example subject</Subject>
      <Document>
        <PlainText>example plain text message</PlainText>
      </Document>
      <Attachment>
        <DocumentName>exampledocument.pdf</DocumentName>
        <File>
          <DocumentType>application/pdf</DocumentType>
          <DocumentData>JVBERi0xLjENMSAwIG9iag08PA0gL1BhZ2VzIDMgMCBSDSAiT3B1bkFjdGlvbiA3IDAgUG0+Pg1lbmRvYmoNMyAwIG9iag08PA0gL1R5cGUgL1BhZ2Vz
          </File>
        </Attachment>
      </ClinicalMessage>
    </Body>
  </N2NMessage>
```

Clinical Message Routed to Receiving System

The following is an example of a single recipient message routed to a receiving system.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<N2NMessage version="Civ6" release="0"
  xmlns="http://www.surescripts.com/messaging">
  <Header>
    <Recipients>
      <To Qualifier="DIRECT" FriendlyName="User Two" Route="true">usertwo@exampledomain.com</To>
    </Recipients>
    <From Qualifier="DIRECT" FriendlyName="User One">userone@exampledomain.com</From>
    <MessageID>S6LMlioObgF10us5fJyBhC6EmZlTTEi</MessageID>
    <SentTime>2018-11-13T15:27:41.184Z</SentTime>
    <SenderSoftware>
      <SenderSoftwareDeveloper>test developer</SenderSoftwareDeveloper>
      <SenderSoftwareProduct>test product</SenderSoftwareProduct>
      <SenderSoftwareVersionRelease>1.0</SenderSoftwareVersionRelease>
    </SenderSoftware>
    <TestMessage />
  </Header>
  <Body>
    <ClinicalMessage>
      <Subject>example subject</Subject>
      <Document>
        <PlainText>example plain text message</PlainText>
      </Document>
    </ClinicalMessage>
  </Body>
</N2NMessage>
```

Status Response – Message Delivered

The following is an example of an HTTP response *Status* message indicating the *Clinical* message was successfully delivered to the recipient.

Bash

```
<?xml version="1.0" encoding="utf-8"?>
<N2NMessage version="Civ6" release="0" xmlns="http://www.surescripts.com/messaging">
  <Header>
    <Recipients>
      <To Qualifier="DIRECT" FriendlyName="User One">userone@exampledomain.com</To>
    </Recipients>
    <From Qualifier="DIRECT" FriendlyName="User Two">usertwo@exampledomain.com</From>
    <MessageID>9eb63133dd41c183309d9af4a7d851</MessageID>
    <RelatesToMessageID>91f6325d9d69422856806cc2d611166</RelatesToMessageID>
    <SentTime>2018-11-13T15:27:41.184Z</SentTime>
    <SenderSoftware>
      <SenderSoftwareDeveloper>test developer</SenderSoftwareDeveloper>
      <SenderSoftwareProduct>test product</SenderSoftwareProduct>
      <SenderSoftwareVersionRelease>1.0</SenderSoftwareVersionRelease>
    </SenderSoftware>
  </Header>
  <Body>
    <Status>
      <Code>010</Code>
    </Status>
  </Body>
</N2NMessage>
```

Status Response – Message Delivered to Multiple Recipients

The following is an example of an HTTP response *Status* message indicating the *Clinical* message was successfully delivered to multiple recipients hosted by a single customer system.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<N2NMessage version="Civ6" release="0" xmlns="http://www.surescripts.com/messaging">
  <Header>
    <Recipients>
      <To Qualifier="DIRECT" FriendlyName="User One">userone@exampledomain.com</To>
    </Recipients>
    <From Qualifier="ZZZ">PortalName</From>
    <MessageID>9eb63133dd41c183309d9af4a7d851</MessageID>
    <RelatesToMessageID>91f6325d9d69422856806cc2d611166</RelatesToMessageID>
    <SentTime>2018-11-13T15:27:41.184Z</SentTime>
    <SenderSoftware>
      <SenderSoftwareDeveloper>test developer</SenderSoftwareDeveloper>
      <SenderSoftwareProduct>test product</SenderSoftwareProduct>
      <SenderSoftwareVersionRelease>1.0</SenderSoftwareVersionRelease>
    </SenderSoftware>
  </Header>
  <Body>
    <Status>
      <Code>010</Code>
    </Status>
  </Body>
</N2NMessage>
```

Verify Message – Message Delivered Electronically

The following is an example of a *Verify* message indicating the *Clinical* message was successfully delivered electronically to the recipient.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<N2NMessage version="CIV6" release="0" xmlns="http://www.surescripts.com/messaging">
  <Header>
    <Recipients>
      <To Qualifier="DIRECT" FriendlyName="User One">userone@exampledomain.com</To>
    </Recipients>
    <From Qualifier="DIRECT" FriendlyName="User Two">usertwo@exampledomain.com</From>
    <MessageID>fbe03a95e68c53240419cbb58236511d</MessageID>
    <RelatesToMessageID>91f6325d9d694222856806cc2d611166</RelatesToMessageID>
    <SentTime>2018-11-13T15:27:41.184Z</SentTime>
    <SenderSoftware>
      <SenderSoftwareDeveloper>test developer</SenderSoftwareDeveloper>
      <SenderSoftwareProduct>test product</SenderSoftwareProduct>
      <SenderSoftwareVersionRelease>1.0</SenderSoftwareVersionRelease>
    </SenderSoftware>
  </Header>
  <Body>
    <Verify>
      <VerifyStatus >
        <Code>010</Code>
      </VerifyStatus>
    </Verify>
  </Body>
</N2NMessage>
```

Error Response – Message Not Delivered

The following is an example of an *Error* response indicating the *Clinical* message could not be delivered to the recipient.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<N2NMessage version="Civ6" release="0" xmlns="http://www.surescripts.com/messaging">
  <Header>
    <Recipients>
      <To Qualifier="DIRECT" FriendlyName="User One">userone@exampledomain.com</To>
    </Recipients>
    <From Qualifier="ZZZ">PortalName</From>
    <MessageID>fbe03a95e68c53240419cbb58236511d</MessageID>
    <RelatesToMessageID>91f6325d9d69422856806cc2d611166</RelatesToMessageID>
    <SentTime>2018-11-13T15:27:41.184Z</SentTime>
    <SenderSoftware>
      <SenderSoftwareDeveloper>test developer</SenderSoftwareDeveloper>
      <SenderSoftwareProduct>test product</SenderSoftwareProduct>
      <SenderSoftwareVersionRelease>1.0</SenderSoftwareVersionRelease>
    </SenderSoftware>
  </Header>
  <Body>
    <Error>
      <Code>600</Code>
    </Error>
  </Body>
</N2NMessage>
```

Error Response – Message Not Delivered to Multiple Recipients

The following is an example of an *Error* response indicating the *Clinical* message could not be delivered to the recipient.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<N2NMessage version="Civ6" release="0" xmlns="http://www.surescripts.com/messaging">
  <Header>
    <Recipients>
      <To Qualifier="DIRECT" FriendlyName="User One">userone@exampledomain.com</To>
    </Recipients>
    <From Qualifier="ZZZ">PortalName</From>
    <MessageID>fbe03a95e68c53240419cbb58236511d</MessageID>
    <RelatesToMessageID>91f6325d9d69422856806cc2d611166</RelatesToMessageID>
    <SentTime>2018-11-13T15:27:41.184Z</SentTime>
    <SenderSoftware>
      <SenderSoftwareDeveloper>test developer</SenderSoftwareDeveloper>
      <SenderSoftwareProduct>test product</SenderSoftwareProduct>
      <SenderSoftwareVersionRelease>1.0</SenderSoftwareVersionRelease>
    </SenderSoftware>
  </Header>
  <Body>
    <Error>
      <Code>600</Code>
    </Error>
  </Body>
</N2NMessage>
```

Example of Namespace in Header

The following is an implementation example defining the cda namespace within the header of the XML.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<N2NMessage version="Civ6" release="0" xmlns:cda="urn:hl7-org:v3" xmlns="http://www.surescripts.com/messaging">
  <Header>
    <Recipients>
      <To Qualifier="DIRECT" FriendlyName="User One">userone@exampledomain.com</To>
    </Recipients>
    <From Qualifier="ZZZ">PortalName</From>
    <MessageID>fbe03a95e68c53240419cbb58236511d</MessageID>
    <RelatesToMessageID>91f6325d9d694222856806cc2d611166</RelatesToMessageID>
    <SentTime>2018-11-13T15:27:41.184Z</SentTime>
    <SenderSoftware>
      <SenderSoftwareDeveloper>test developer</SenderSoftwareDeveloper>
      <SenderSoftwareProduct>test product</SenderSoftwareProduct>
      <SenderSoftwareVersionRelease>1.0</SenderSoftwareVersionRelease>
    </SenderSoftware>
  </Header>
  <Body>
    <ClinicalMessage>
      <Subject>Test Message for ERX-24414</Subject>
      <Document>
        <PlainText/>
      </Document>
      <Attachment>
        <DocumentName>Latifa.CDA.xml</DocumentName>
        <cda:ClinicalDocument>
          <cda:realmCode code="US" />
          <cda:typeId root="2.16.840.1.113883.1.3" extension="POCD_HD000040" />
          <cda:templateId root="2.16.840.1.113883.10.20.22.1.1" />
          <cda:templateId root="2.16.840.1.113883.10.20.22.1.2" />
          <cda:id root="2.16.840.1.113883.3.2054" extension="7f2f1a44ab934cddave108a43285e7b7" />
          <cda:code code="34133-9" displayName="Summarization of episode note" codeSystem="2.16.840.1.113883.6.1" codeSystemName="
          <cda:title>Dev Imm Title</cda:title>
          <cda:effectiveTime value="20130903082505" />
          <cda:confidentialityCode code="N" displayName="Normal" codeSystem="2.16.840.1.113883.5.25" codeSystemName="HL7 Confidential
          <cda:languageCode code="US" />
          <cda:documentationOf>
            <cda:serviceEvent classCode="PCPR">
              <cda:effectiveTime>
                <cda:low nullFlavor="UNK" />
                <cda:high nullFlavor="UNK"/>
              </cda:effectiveTime>
            </cda:serviceEvent>
          </cda:documentationOf>
        </cda:ClinicalDocument>
      </Attachment>
    </ClinicalMessage>
  </Body>
</N2NMessage>
```

Example of Namespace in Element

The following is an implementation example defining the cda namespace within the ClinicalDocument element of the XML.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<N2NMessage version="Civ6" release="0" xmlns:cda="urn:hl7-org:v3" xmlns="http://www.surescripts.com/messaging">
  <Header>
    <Recipients>
      <To Qualifier="DIRECT" FriendlyName="User One">userone@exampledomain.com</To>
    </Recipients>
    <From Qualifier="ZZZ">PortalName</From>
    <MessageID>fbc03a95e68c53240419cbb58236511d</MessageID>
    <RelatesToMessageID>91f6325d9d69422856806cc2d611166</RelatesToMessageID>
    <SentTime>2018-11-13T15:27:41.184Z</SentTime>
    <SenderSoftware>
      <SenderSoftwareDeveloper>test developer</SenderSoftwareDeveloper>
      <SenderSoftwareProduct>test product</SenderSoftwareProduct>
      <SenderSoftwareVersionRelease>1.0</SenderSoftwareVersionRelease>
    </SenderSoftware>
  </Header>
  <Body>
    <ClinicalMessage>
      <Subject>Test Message for ERX-24414</Subject>
      <Document>
        <PlainText />
      </Document>
      <Attachment>
        <DocumentName>Latifa.CDA.xml</DocumentName>
        <cda:ClinicalDocument xmlns="urn:hl7-org:v3" xmlns:xsltUtil="urn:xsltUtil">
          <cda:realmCode code="US" />
          <cda:typeId root="2.16.840.1.113883.1.3" extension="POCD_HD000040" />
          <cda:templateId root="2.16.840.1.113883.10.20.22.1.1" />
          <cda:templateId root="2.16.840.1.113883.10.20.22.1.2" />
          <cda:id root="2.16.840.1.113883.3.2054" extension="7f2f1a44ab934cddave108a43285e7b7" />
          <cda:code code="34133-9" displayName="Summarization of episode note" codeSystem="2.16.840.1.113883.6.1" codeSystemName="
          <cda:title>Dev Imm Title</title>
          <cda:effectiveTime value="20130903082505" />
          <cda:confidentialityCode code="N" displayName="Normal" codeSystem="2.16.840.1.113883.5.25" codeSystemName="HL7 Confident
          <cda:languageCode code="US" />
          <cda:documentationOf>
            <cda:serviceEvent classCode="PCPR">
              <cda:effectiveTime>
                <cda:low nullFlavor="UNK" />
                <cda:high nullFlavor="UNK" />
              </cda:effectiveTime>
            </cda:serviceEvent>
          </cda:documentationOf>
        </cda:ClinicalDocument>
      </Attachment>
    </ClinicalMessage>
  </Body>
</N2NMessage>
```

7. Clinical Direct Messaging Processing Summary

This section explains the processing that takes place for *Clinical* messages exchanged between a sending customer and a receiver.

Event ID	Event	Process	Type of Message	Code	Desc. Code	Description
1.0	Surescripts receives a message					
1.1		Surescripts fails to recognize the type of message.	N/A	N/A	N/A	A text message that says "Malformed payload sent to system"
1.2		Surescripts recognizes that the message is a clinical message but does not have enough information to create an error message.	N/A	N/A	N/A	A text message that says "Malformed payload sent to system"
1.3		Sender ParticipantID or Password is Invalid	N/A	N/A	N/A	A text message that says "Authentication failed: Invalid UID and password." Or "Missing authorization info in the header"
1.4		Any other system-related error which makes it impossible for Surescripts to accept the message.	N/A	N/A	N/A	A general text message describing the error.
2.0	Surescripts processes the message					
2.1		Surescripts determines that the sender is not valid.	Error	900 – Transaction rejected	001 – Sender ID not on file	
2.2		Surescripts determines that the sender is not certified to send this type of message.	Error	900 – Transaction rejected	None	Message type not supported by sender
2.3		Surescripts determines that the receiver is not valid.	Error	900 – Transaction rejected	002 – Receiver ID not on file.	
3.0	Synchronous Message Delivery					
3.1		Message was delivered to recipient.	Status	010	Specified by receiving system	

Event ID	Event	Process	Type of Message	Code	Desc. Code	Description
3.2		Message could not be delivered to recipient.	Error	601	Specified by the receiving system	
3.3		A problem processing the message was encountered by the receiving system.	Error	602	Specified by the receiving system	
4.0	Surescripts accepts message for delivery					
4.1		Surescripts sends a Status message to sender of clinical message, indicating it was accepted for delivery.	Status	000	None	
4.1.0	Surescripts attempts delivery	The result of the delivery attempt will be communicated back to the sender by means of one of the following free standing messages.				
4.1.1		Message is delivered to the recipient.	Verify	010	None	
4.1.2		Message could not be delivered to recipient.	Error	601	Specified by the receiving system.	
4.1.3		A problem processing the message was encountered by the receiving system.	Error	602	Specified by the receiving system.	

8. Changes From Previous Versions

Changes from Version 1.5 to 6.0

- Updated header to include version="C1v6" release="0"
- Changes to message header elements:
 - Increased MessageID length from ..35 to ..255
 - Increased RelatesToMessageID length from an..35 to an..255

Changes from Version 1.4 to 1.5

- Updated header to include version="C1v1" release="5"
- Added the following conditional elements to the Message header:
 - To/@FriendlyName
 - To/@Route
 - From/@FriendlyName

Changes from Version 1.3 to 1.4

- Updated header to include version="C1v1" and release="4".
- Added multiple-recipient messaging.

Changes from Version 1.2 to 1.3

- Updated header to include version="C1v1" and release="3".
- Added Patient Demographics section to the header. Previous versions used "x-patient" document data items.
- Removed PlainText and HtmlText attachment types. These are no longer discrete attachment types; however, this content can still be sent as a Base64 encoded File attachment.
- Removed Network Address from RecipientInformation
- Removed unused items from the schema. "HighestVersionSupported" attribute and "Mailbox", "Security", and "ClinicalEvent/MessageRead" elements were removed from the schema.
- Increased DocumentName length from 35 to 255 characters.

Changes from Version 1.1 to 1.2

- Added DIRECT qualifier on From and To elements. This allows messages to be sent to/from trading partners using a direct address
- Recipients collection in the header changed to single Recipient element to reflect that only one (1) recipient is supported.
- Moved Status, Error and Verify messages to the N2NMessage body to allow for routing with the DIRECT qualifier.

9. Disclaimers

Guide Disclaimer

In the event that the Participant chooses to make any software changes based upon recommendations in this guide, the Participant acknowledges and agrees that Surescripts Network shall bear no responsibility or liability for the Participant's changes or any effects thereof. The Participant shall also be required to transition to the new guide at such time as said guide is published, which may involve different or additional parameters than are published in this guide.

Examples Disclaimer

Examples provided throughout this guide are not intended to be all-inclusive. This pertains to example workflows, element-specific (field) examples, or message examples. Participants should not restrict coding to the examples used herein.

When developing, this guide should be used along with additional documentation referenced and applicable customary coding standards.

Copyright

Copyright© 2026 by Surescripts, LLC. All rights reserved.

This document and the information contained within are the intellectual property of Surescripts, LLC. While Surescripts encourages the sharing of this information for knowledge and learning purposes, unauthorized reproduction, distribution, or modification of this material is prohibited.

For sections of this Customer Guide pertaining to NCPDP Standards:

Materials Reproduced With the Consent of ©National Council for Prescription Drug Programs, Inc.

NCPDP is a registered trademark of the National Council for Prescription Drug Programs, Inc. Membership of NCPDP acknowledges your support and commitment to implement these standards as specified in the standards documentation.

You further acknowledge that the Council's standards documents and their predecessors include proprietary material protected under the U.S. Copyright Law and that all rights remain with NCPDP.

10. Document Change Log

The table below tracks significant changes made to the document since it was last published.

▼ 2026-08-21

Section Title	Change Description
N/A	N/A - Developer Portal - initial release