NCPDP XML STANDARD

VERSION 2Ø13Ø71

The document is the basis for XML-based transactions used in NCPDP and supports the general transactions that may be used in implementation guides.

JULY 2Ø13

National Council for Prescription Drug Programs
924Ø East Raintree Drive
Scottsdale, AZ 8526Ø

Phone: (48Ø) 477-1ØØØ
Fax: (48Ø) 767-1Ø42
E-mail: ncpdp@ncpdp.org
http: www.ncpdp.org

NCPDP XML Standard

Version 2Ø13Ø71

COPYRIGHT (©) National Council for Prescription Drug Programs, Inc.
2Ø13

NCPDP recognizes the confidentiality of certain information exchanged electronically through the use of its standards. Users should be familiar with the federal, state, and local laws, regulations and codes requiring confidentiality of this information and should utilize the standards accordingly.

NOTICE: In addition, this NCPDP Standard contains certain data fields and elements that may be completed by users with the proprietary information of third parties. The use and distribution of third parties' proprietary information without such third parties' consent, or the execution of a license or other agreement with such third party, could subject the user to numerous legal claims. **All users are encouraged to contact such third parties to determine whether such information is proprietary and if necessary, to consult with legal counsel to make arrangements for the use and distribution of such proprietary information**.

Published by:
National Council for Prescription Drug Programs
Publication History:
Version 2Ø1Ø121 December 2Ø1Ø
Version 2Ø11Ø71 July 2Ø11
Version 2Ø12Ø31 March 2Ø12
Version 2Ø13Ø11 January 2Ø13
Version 2Ø13Ø41 April 2Ø13
Version 2Ø13Ø71 July 2Ø13

TABLE OF CONTENTS

1. INTRODUCTION

This NCPDP ***XML Standard*** is intended to provide implementation guidance to vendors on the NCPDP XML-based transactions. The document is the basis for XML-based transactions used in NCPDP and supports the general transactions that may be used in implementation guides.

If you have any questions regarding the availability or content of the NCPDP ***XML Standard***, see www.ncpdp.org, or contact the Council office at (48Ø) 477-1ØØØ or via e-mail at ncpdp@ncpdp.org.

1.1 DOCUMENT SCOPE

This document contains the specification. Users of this document should consult the NCPDP documents listed below for further information and clarification.

SCHEMAS USED

This package of information contains the actual XML schema use in implementing NCPDP XML transactions. Schemas include

transport.xsd	The NCPDP transport.xsd defines the envelope structures for the transactions and highest level elements for each transaction.
structures.xsd	The NCPDP structures.xsd defines common and resuable domain structures typically composed of datatypes and potentially other structures.
ecl.xsd	Vocabulary constraints for a business transaction.
datatypes.xsd	The NCPDP datatypes.xsd defines small, static, reusable structures (structural components) whose usage and validation is not typically impacted by the context of it use. For instance, PostalCode structure and validation is not impacted if it used as part of a pharmacy address or a patient's address.
Transaction schemas	*(examples but not limited to)*
script.xsd	The NCPDP script.xsd defines SCRIPT domain transactions
pa-structures.xsd	The NCPDP pa-structures.xsd defines domain structures typically composed of datatypes and potentially other structures which are specific to prior authorization functions.
specialized.xsd	The NCPDP specialized.xsd defines Business Domain Transactions.

NCPDP SCRIPT STANDARD IMPLEMENTATION GUIDE

This document contains the general information needed for implementing NCPDP XML transactions used in electronic prescribing.

NCPDP SPECIALIZED STANDARD IMPLEMENTATION GUIDE

This document contains the general information needed for implementing NCPDP XML transactions used in other exchanges.

DATA DICTIONARY

Full reference to all fields and values (contained within or reference to the *External Code List*) used in the NCPDP standard with examples.

EXTERNAL CODE LIST

Full reference to values used in the NCPDP standard.

STANDARDS MATRIX

This document contains a high-level overview of the latest version/release and/or the most commonly used of those standards and implementation guides, as well as NCPDP's Data Dictionary and External Code List. Additionally, this document provides version/release/publication reference charts for approved and draft NCPDP standards/implementation guides.

These documents are available with NCPDP membership; contact the NCPDP office at 48Ø-477-1ØØØ, or via Internet e-mail at ncpdp@ncpdp.org. The documents are available in the "Members" section of the website at www.ncpdp.org.

2. BACKGROUND

This document provides general guidelines for developers of systems who wish to provide business functionality of NCPDP-based XML transactions to their clients. The document is the basis for XML-based transactions used in NCPDP and supports the general transactions that may be used in implementation guides.

3. XML IMPLEMENTATION

This section provides details and guidelines for developers of systems to exchange transactions utilizing an XML implementation. This section describes the XML standard, and other variables related to the use of an XML implementation of NCPDP. Knowledge of XML is presumed.

3.1 ADDITIONAL INFORMATION SOURCES

3.1.1 XML AND XSD

The XML standard used is defined and maintained by World Wide Web Consortium (W3C) and can be found at http://www.w3.org/TR/REC-xml

This XML schema, also a W3C standard, is used to define the XML standard structure. Its description can be found at http://www.w3.org/XML/Schema#dev

XML Schemas are documents that are used to define and validate the content and structure of XML data, just as a database schema defines and validates the tables, columns, and data types that make up a database. An XML Schema defines and describes certain types of XML data by using the XML Schema definition language (XSD). XML Schema elements (elements, attributes, types, and groups) are used to define the valid structure, valid data content, and relationships of certain types of XML data. XML Schemas can also provide default values for attributes, and elements.

XML Schema is used to guarantee consistency among certain types of XML data that is shared between applications and organizations. Organizations and applications that wish to exchange data can build their applications around these schemas so their xml transactions will be understood. For example, a prescription represented in XML can be validated with an XML Schema before it is sent between business partners. This validation verifies that all of the elements (individual pieces) of data exist, are in the expected sequence, and are all of the correct data type. This ensures that the recipient of the prescription will be able to interpret the data correctly when it is received. (Reference: www.msdn.microsoft.com, “Introduction to XML Schemas”.)

4. BUSINESS ENVIRONMENT

NCPDP ***XML Standard*** is a data transmission intended to facilitate the communication of information between prescribers/providers, pharmacies, payers, other entities, and their trading partners.

4.1 PARTICIPANTS

The entities that may participate in the NCPDP ***XML Standard*** exchange of information include, but are not limited to, Pharmacies, Prescribers, Payers, Facilities, Value Added Networks/Switches/Intermediaries. Specific roles and participation may be affected by State and Federal laws.

A "PROVIDER" is a licensed entity that dispenses prescription drugs and provides professional pharmacy services, such as clinical pharmacy services (consulting) respective to the dispensing function. The entity may be a retail/chain, mail order, or independent pharmacy, prescriber, hospital, or long term care facility. A 'PROVIDER' may be a retail pharmacy, mail order pharmacy, doctor's office, clinic, hospital, long-term care facility, or any other entity which dispenses prescription drugs and submits those prescriptions to a payer for reimbursement.

A "PRESCRIBER" is a licensed entity that prescribes prescription drugs and provides professional medical services, such as clinical services respective to the prescribing function. The entity may be a clinic or independent prescriber, hospital, or care facility.

A "SWITCH/INTERMEDIARY" is an entity that accepts an electronic transaction from another organization and electronically routes the transaction to a receiving entity. For example, a pharmacy can send requests to many prescribers and a prescriber can send requests to many pharmacies. An intermediary/switch may be required to shield the participants from the uniqueness of the other parties.

A "PAYER" is often a third-party administrator of prescription drug programs on behalf of insurers. The payer also may be an insurer, a governmental program or any other entity which receives prescription drug claims.

A "CENTRAL FILL FACILITY" is a pharmacy permitted to fill prescriptions on behalf of another pharmacy based on common ownership or contractual relationship providing for dispensing activity services. Prescription information is electronically transmitted to the central fill pharmacy and fulfillment is based on the electronic record in accordance with all applicable regulations and requirements.

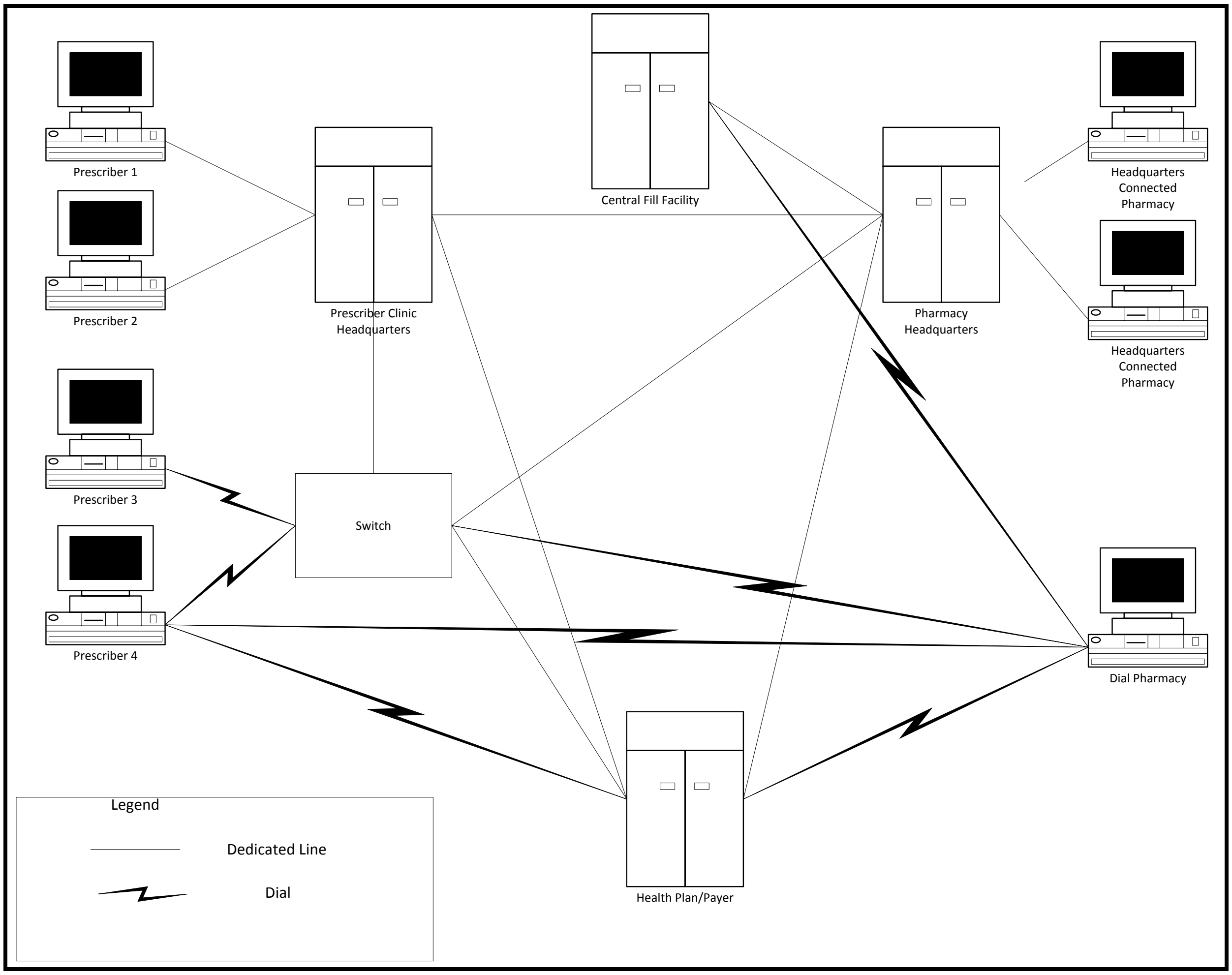


Figure 1. Connection of NCPDP XML Exchange Parties

4.2 RESPONSIBILITIES OF THE PARTICIPANTS

When using this standard, the Sender, Switch, and the Receiver are expected to perform specific functions, as outlined below:

4.2.1 RESPONSIBILITIES OF THE SENDER

At a high-level, the Sender is responsible for:

- Populating all mandatory fields for this request transmission.
- Establishing the connection with the Switch or Receiver, and initiating the telecommunication session.
- Formatting the request and sending it in the transaction envelope that is appropriate to the protocol being used.
- Interpreting and acting upon any response provided by the Receiver. This could vary from Receiver to Receiver and from time to time (i.e. during an equipment problem). This could also include the situation where no response is received.
- Terminating the session and disconnecting the transmission.

4.2.2 RESPONSIBILITIES OF THE SWITCH

At a high-level, the Switch is responsible for:

- Establishing the connection with the Receiver and delivering the request from the Sender.
- Interpreting the request submitted by the Sender and responding as needed to provide the maximum amount of information for error correction and resolution when required.
- Providing the ability to convert versions of the standard as feasibly possible and needed based on trading partner agreements.
- Returning the response from the Receiver to the Sender.
- Providing a high level of system availability and providing a viable fall-back mechanism in the event of equipment failure.

4.2.3 RESPONSIBILITIES OF THE RECEIVER

At a high-level, the Receiver is responsible for:

- Interpreting requests submitted by the Sender and responding as needed to provide the maximum amount of information for error correction and resolution when required.
- Populating all mandatory fields for this response transmission.
- Formatting the response and sending it in the transaction envelope that is appropriate to the protocol being used.
- Ignoring irrelevant allowed data that may be supplied by the Originator (i.e. the request may have data in fields not required for a particular receiver. This should not create an error.)
- Recognizing and supporting multiple versions of the standard for a long enough period of time to allow the users to convert their processing as new versions of the standard are developed and released.
- Providing a high level of system availability and providing a viable fall-back mechanism in the event of equipment failure.

4.3 COMMUNICATION MODES

- DIRECT CONNECT - where there is a direct connection between participating entities.
- MAILBOX - where each business partner retrieves transactions that are placed in their mailbox.
- COMBINATION - where communication modes may include a combination of Mailbox and Direct Connect among business partners.

In each communication mode, a switch may act as an intermediary business partner to certify transactions, deliver transactions and manage mailboxes.

4.3.1 DIRECT CONNECT

In **Figure 1**, the connection between the prescriber clinic headquarters and the pharmacy headquarters, is illustrative of a direct connect (as in a dedicated line). In this business model, any transactions destined for either prescriber 1 or 2, or pharmacy 1 or 2, can be delivered immediately upon sending.

4.3.2 MAILBOX

In **Figure 1**, the prescriber's and pharmacy's systems may connect to a Mailbox via a switched network, as in a dial out scenario. This may be represented by the path of Prescriber 3 or 4 to the Switch, or the path of the Dial Pharmacy to Switch. The Switch may perform the role of a Mailbox by storing transactions which cannot be delivered immediately to a destination which functions in dial modes. The Switch may also perform the role of a communications link between parties.

4.3.3 COMBINATION

As shown in **Figure 1**, the business model might be represented by the path between Prescriber 3, the Switch, and the Pharmacy Headquarters. In this scenario, any transactions sent by the Prescriber can be immediately delivered to the Pharmacy Headquarters via the Switch having a direct connection to the Pharmacy Headquarters. Transactions sent from the Pharmacy to the Prescriber would be stored in the Switch's mailbox until the prescriber's system either dials in to pick up mail, or the Mailbox dials out to the Prescriber's system.

4.4 DIAL SCENARIOS

This implementation can be used by pharmacy or prescriber systems to send transactions in different dial scenarios. The choice of scenario is based on the trading partner agreements and business needs. These scenarios are dial up and dial in. Dial up is the commonly used scenario where the pharmacy or prescriber system places a call to the Mailbox and establishes a connection. Dial in is the scenario where the Mailbox calls the pharmacy or prescriber system and connects.

Since the prescriber or pharmacy may not be directly connected to Switch, transactions received from prescribers or pharmacies are stored in a “mailbox” on the Switch's system. Since these transactions can arrive at the Switch without the knowledge of the pharmacy or prescriber system, there may be two strategies to deliver this mail. The first is to await the connection to the Mailbox by the pharmacy or prescriber system. The second involves the Mailbox calling the pharmacy or prescriber system when transactions are received. The former requires that the pharmacy or prescriber system call the Mailbox periodically even when it has nothing to send. The latter relies on the Mailbox to call the pharmacy or prescriber system when mail has arrived and the pharmacy or prescriber system only calls the Mailbox when it has transactions to initiate.

In the dial in scenario, the Mailbox will dial the pharmacy or prescriber system to notify of mail. The Mailbox will establish a connection to the system, using any appropriate log on and sync up requirements. The pharmacy or prescriber system accepts the dial in call. The prescriber or pharmacy system will then generate a GetMessage. The Mailbox will respond with the first piece of mail. The prescriber or pharmacy system initiates a second GetMessage and the Mailbox returns the next piece of mail (if there is one). This conversation continues until the Mailbox responds with a Status transaction of No More Mail.

In the dial up scenario, the Mailbox may not be able to deliver the transaction until the prescriber or pharmacy system calls in to retrieve transactions. In the dial in scenario, the Mailbox may call the pharmacy or prescriber system, but the time at which this can successfully occur can be affected by a number of parameters. Even if the Mailbox can deliver the request to the pharmacy's or prescriber's system quickly, the pharmacy or prescriber may not make a decision for some time. As with the phone calls which occur today to accomplish the same effect, the prescriber may only make such decisions a few times a day, or during business hours, et cetera.

The dial prescriber or pharmacy system is in control of the connection in these scenarios. It may continue to perform functions until it has completed its work. However, the Mailbox will not allow a dial pharmacy or prescriber system to remain connected without performing transactions.

Therefore, the pharmacy and prescriber system should be designed to prepare all activity it expects to perform with the Mailbox, place the call, perform that activity and disconnect.

4.5 IDENTIFIERS

4.5.1 PHARMACY

A common standard identifier is the NCPDP Provider ID Number to identify a pharmacy. Every pharmacy has a unique NCPDP Provider ID Number assigned to them. The highest level of routing is the pharmacy NCPDP Provider ID Number in the <To> or <From> fields. Routing to individual systems within a pharmacy is at the discretion of the software system. Likewise, when routing to a pharmacy or clinic within a "chain" setting, the highest level identifier is still the pharmacy NCPDP Provider ID Number. It is the responsibility of the "hops" in between (switches, headquarters) to recognize the relationship between pharmacy/prescriber and the "chain" and send the transactions down the appropriate route. <TertiaryIdentifier> may be a chain designation, a switch, or clinic designation, at trading partner agreements.

4.5.2 PRESCRIBER

A Mailbox may elect to assign identifiers to each prescriber's system, as part of the enrollment service. Note: It is the **system** and not the prescriber that is identified at the transmission level in the <To> or <From> fields. The highest level of routing is the clinic. Routing to individual prescribers (or systems) is at the discretion of the software system.

4.5.3 MAILBOX

The Mailbox will also have an identifier if used within trading partners. This identifier is used for transactions that are addressed specifically to the Mailbox (i.e., getting mail, password changes).

4.5.4 PAYERS

Currently there is no standard for the identification of a payer. In pharmacy claims processing the BIN is commonly used to identify a payer. The Department of Health and Human Services is expected to publish a proposed rule for a National Health Plan ID, but this must go through the federal rule making process to be finalized. It is unclear if this ID will be useful in the electronic prescribing environment to correctly identify the payer entity that has medication history information. In lieu of a standard identifier, <To> or <From> fields when used to identify a payer will contain the qualifier value of “ZZ” (Mutually Defined) and the defined value between trading partners.

4.6 DIRECTORY SERVICES

It will be necessary for the pharmacy and prescriber system to provide a mechanism to store and find the appropriate ID. For example, a “pick” list could be provided. This list may be manually updateable by the pharmacy or prescriber system as no standards currently exist to support the distribution of directory information electronically.

4.7 FAILURE CONSIDERATIONS

4.7.1 DELAYS

It is important to understand that the Mailbox will attempt to avoid any delays in the delivery of the prescription. However, dedicated link failures, mail not picked up timely, communication problems, etc. can result in the late arrival of a prescription. The pharmacy headquarters system or the pharmacy management system needs to be capable of managing scripts that have arrived “too late.” To further expand on this problem, it is possible that due to a communication or line outage, the Mailbox is unable to deliver a new prescription to the pharmacy timely. In the meantime, the patient has arrived at the pharmacy; the pharmacy has called the prescriber, and filled the script. The patient has left the pharmacy with the prescription filled. The communication problem is resolved, so the Mailbox is now able to send the new prescription to the pharmacy. How the pharmacy headquarters system or pharmacy management system handles this case is at the discretion of the software developer, but this case should not be ignored. Please note that the rate of occurrence of this delay should be very low, but it can occur.

5. BUSINESS FUNCTIONS

These operational functions are supported.

- GetMessage - The GetMessage is used to manage the delivery of transactions that have arrived at Mailbox while the pharmacy or prescriber system was not connected. These are referred to as mailboxed transactions.
- Verify - The Verify serves to inform the sender of a transaction that the transaction has been delivered to the receiver. Logically, it can be described as the "receipt" portion of a "return receipt requested" transaction.
- PasswordChange - To validate the source of all transactions, the sender may be required to supply a password. This function allows the sender to change their password. A Mailbox may require that the very first transaction from a prescriber or pharmacy system establish the password. An example of this is given in the "*Transmission Examples*" section of this document.

5.1 TRANSACTION TYPES

The transaction types covered in this guide include

GetMessage This is the transaction used by the prescriber or pharmacy asking the mailbox if there are any transactions.

PasswordChange This is the transaction used to request a password change.

Verify If the sender asked for verification that the recipient did in fact receive the transaction, this is the transaction type that is sent back to the sender.

Status Is used to relay acceptance of a transaction back to the sender. A Status in response to RefillRequest, RefillResponse, NewRx, Verify, Error, RxChangeRequest, RxChangeResponse, CancelRx, CancelRxResponse, RxFill, MTMServiceRequest, MTMServiceResponse, NewRxRequest, NewRxResponseDenied, ClinicalInfoRequest or ClinicalInfoResponse indicates acceptance and responsibility for a request. A Status in response to GetMessage indicates that no mail is waiting for pickup. A Status cannot be mailboxed and may not contain an error.

Error This indicates an error has occurred indicating the request was terminated. (An Error can be generated when there is a communication problem or when the transaction actually had an error.)

5.1.1 QUERY INTRODUCTION

Query transactions are used for the exchange of patient-centric clinical health information, such as allergies, conditions and medical histories between electronically enabled healthcare providers (see the NCPDP ***Specialized Standard Implementation Guide***). Query transactions are also used by a pharmacy to request a NewRx prescription from a prescriber (see the NCPDP ***SCRIPT Standard Implementation Guide***). It is recommended that all functions of the query be supported.

- NewRxRequest
- NewRxResponseDenied
- ClinicalInfoRequest
- ClinicalInfoResponse

6. DISCUSSION OF STATUS, ERROR, AND VERIFY TRANSACTIONS

Before expanding on the nature of the transactions, it is necessary to discuss the transactions exchanged to implement these transactions. Each of the transactions discussed below consists of a transaction related to the corresponding transaction and a response sent by the receiver of the transaction. The nature of that response consists of either a Status, an Error, or in certain cases a Verify. Status response transactions always indicate that all is well and, correspondingly, Error response transactions indicate that there is a problem with the transaction. The system supports a Verify transaction as a response under certain circumstances described below.

6.1 STATUS

The Status response is used to relay acceptance of a transaction back to the sender. A Status in response to any applicable transaction other than GetMessage indicates acceptance and responsibility for a request. A Status in response to GetMessage indicates that no mail is waiting for pickup. **A Status cannot be mailboxed and may not contain an error.** Therefore, a Status can be used to reply to the following:

- RefillRequest
- RefillResponse
- NewRx
- RxChangeRequest
- RxChangeResponse
- RxFill
- CancelRx
- CancelRxResponse
- GetMessage (no more mail)
- PasswordChange
- Verify
- Error
- RxHistoryRequest
- RxHistoryResponse
- Census
- Resupply
- DrugAdministration
- MTMServiceRequest
- MTMServiceResponse
- NewRxRequest
- NewRxResponseDenied
- ClinicalInfoRequest
- ClinicalInfoResponse
- PAInitiationRequest
- PAInitiationResponse
- PARequest
- PAResponse
- PAAppealRequest
- PAAppealResponse
- PACancelRequest
- PACancelResponse
- CFInventoryList
- CFProductInquiry
- CFProductInquiryResponse
- CFRxOrderRequest
- CFRxOrderCompletion
- CFRxOrderCancel
- CFManifest

The response can be returned by either the pharmacy/prescriber headquarters system (in dedicated line scenarios) or the pharmacy/prescriber management system. **It cannot be responded to by both.** The pharmacy/prescriber management system is the preferred responder, if this is consistent with the pharmacy/prescriber architecture.

6.2 ERROR

Error responses always indicate that there is a problem with the transaction. An Error can be mailboxed, as it may be signifying to the originator that a transaction was unable to be delivered or encountered problems in the acceptance. The Error must be a different response than a Status, since the communication between the prescriber or pharmacy system and the Mailbox must clearly denote the actions taking place. An Error is a response being delivered on behalf of a previous transaction, and the Status signifies no more mail. The Error response can be used to reply to the following:

- RefillRequest
- RefillResponse
- NewRx
- RxChangeRequest
- RxChangeResponse
- CancelRx
- CancelRxResponse
- RxFill
- GetMessage
- PasswordChange
- Verify
- RxHistoryRequest
- RxHistoryResponse
- Census
- Resupply
- DrugAdministration
- MTMServiceRequest
- MTMServiceResponse
- NewRxRequest
- NewRxResponseDenied
- ClinicalInfoRequest
- ClinicalInfoResponse
- PAInitiationRequest
- PAInitiationResponse
- PARequest
- PAResponse
- PAAppealRequest
- PAAppealResponse
- PACancelRequest
- PACancelResponse
- CFInventoryList
- CFProductInquiry
- CFProductInquiryResponse
- CFRxOrderRequest
- CFRxOrderCompletion
- CFRxOrderCancel
- CFManifest

6.3 VERIFY

Verifications result when a “return receipt requested” flag is set in the original request. The system supports the setting of this flag in NewRx transactions, RxChangeRequest and RxChangeResponse transactions, RxFill, RefillRequest and RefillResponse, MTMServiceRequest and MTMServiceResponse, NewRxRequest and NewRxResponseDenied, and ClinicalInfoRequest and ClinicalInfoResponse transactions. Upon receiving a transaction with ReturnReceipt set, it is the responsibility of the receiver to either

- Generate a Verify in response to the request, or
- Generate a Status in response to this request, followed subsequently by a “free standing” Verify.

It is recommended that the first bullet method above is utilized. The Verify transaction should be sent when the software system has received the initial request with return receipt requested. The Verify is notifying the originator that the transaction was received at the software system. It is not a notification of action taking place, since time may elapse before the ultimate answer to the transaction may take place. The “*Verify Transaction”* and “*Transmission Examples*” sections should help to clarify this requirement.

The Verify transaction may be used to respond to the following, when the <ReturnReceipt> field is set in the original transaction:

- RefillRequest
- RefillResponse
- NewRx
- RxChangeRequest
- RxChangeResponse
- RxFill
- CancelRx
- CancelRxResponse
- RxHistoryRequest
- RxHistoryResponse
- Census
- Resupply
- DrugAdministration
- MTMServiceRequest
- MTMServiceResponse
- NewRxRequest
- NewRxResponseDenied
- ClinicalInfoRequest
- ClinicalInfoResponse
- PAInitiationRequest
- PAInitiationResponse
- PARequest
- PAResponse
- PAAppealRequest
- PAAppealResponse
- PACancelRequest

- PACancelResponse
- CFInventoryList
- CFProductInquiry
- CFProductInquiryResponse
- CFRxOrderRequest
- CFRxOrderCompletion
- CFRxOrderCancel
- CFManifest

7. TRANSACTIONS

7.1 GETMESSAGE TRANSACTION

The GetMessage transaction is at the heart of the mechanism used by a pharmacy or prescriber system to receive transactions from each other via a Switch, acting as a Mailbox. Since the prescriber or pharmacy may not be directly connected to a Switch, transactions received from prescribers or pharmacies are stored in a “mailbox” on the Switch’s system. Since these transactions can arrive at the Switch without the knowledge of the pharmacy or prescriber system, there may be two strategies to deliver this mail. The first is to await the connection to the Mailbox by the pharmacy or prescriber system. The second involves the Mailbox calling the pharmacy or prescriber system when transactions are received. The former requires that the pharmacy or prescriber system call the Mailbox periodically even when it has nothing to send. The latter relies on the Mailbox to call the pharmacy or prescriber system when mail has arrived and the pharmacy or prescriber system only calls the Mailbox when it has transactions to initiate. In either case, once the connection is made, the mail is removed by the pharmacy or prescriber system by issuing GetMessage to Mailbox. The response to each GetMessage request will be one of the following.

Pharmacy System:

- NewRx - A new prescription transaction represents a prescription for a patient.
- RxChangeResponse - A prescription change response is the response to a previously sent RxChangeRequest. It can approve the requested change, with or without reasons.
- RefillResponse - A refill prescription response is the response to a previously sent RefillRequest. It can approve the requested change, with or without comments, or it can decline the requested change, with or without reasons.
- CancelRx - A cancel prescription request indicates the desire on the part of the provider to cancel a prescription already transmitted to the pharmacy.
- Long Term Care (LTC) Medication Change - A prescriber has the need to modify the order and notify the pharmacy. The prescriber system will always send a CancelRx and a NewRx, regardless of the type of change, to the pharmacy.
- Verify - This verification is the response to a RefillRequest, RxFill, RxChangeRequest, or CancelRxResponse sent previously with a return receipt requested by the pharmacy system. It indicates that the prescriber has received the request.
- Census - originated by the facility in a long term care environment. The transaction notifies the pharmacy about census events. The transaction can be used in three cases - to notify the pharmacy of a new resident, a change to demographic information of a resident, or the discharge of a resident.
- Resupply - Request to send a refill from a facility to a pharmacy. An example use case is when a medication supply for a resident is running low (2-3 doses) and a new supply is needed from the pharmacy, the nurse needs a way to notify the pharmacy that a refill for the medication is needed.
- MTMServiceRequest – A processor is requesting the pharmacy provide a service to a patient.
- NewRxResponseDenied – A denied response to a previously sent NewRxRequest. (If approved, a NewRx would be sent.)
- ClinicalInfoRequest - A request between healthcare providers for patient-centric clinical information.
- ClinicalInfoResponse – A response to the ClinicalInfoRequest to indicate if the information will/can be provided. The ClinicalInfoResponse can contain an attachment of the Clinical Information.
- Error - Errors can be received under two different conditions. It is necessary to examine the recipient in the <To> field and <MessageID> field to determine its meaning. If the source is Mailbox and the trace number refers to the just sent GetMessage, then an error has been discovered by Mailbox in the GetMessage request itself. Otherwise, it refers to a previously

sent transaction that was found to be in error by the prescriber's system.

- Status - A status in response to a GetMessage indicates that no more mail is available. Indirectly, then, any response to a GetMessage request other than a Status implies that additional mail *may* be in the mailbox and, therefore, solicits another GetMessage. A Status transaction is never mailboxed and cannot contain an error.

Prescriber:

- RefillRequest - A refill prescription request indicates the desire on the part of the pharmacy to dispense additional drugs based on a previously obtained prescription which has expired due to date or number of refills allowed. The prescriber can approve the requested change, with or without comments, or decline the requested change, with or without reasons and so indicate by sending a RefillResponse to the pharmacy.
- RxChangeRequest - A prescription change request indicates the desire on the part of the pharmacy to alter or clarify a new prescription or an existing "fillable" prescription. The pharmacy may request a substitution, alert of a therapeutic or drug utilization review interchange, which may require the change of a new prescription recently processed. It may also be utilized to request a prescriber to review the drug requested, and obtain a prior authorization from the payer for the prescription.
- RxFill - A prescription fill status notification indicates the desire on the part of the pharmacy to notify the prescriber on the fill status of a prescription. It can inform the prescriber that the prescription has been dispensed, not dispensed, or partially dispensed.
- CancelRxResponse - A cancel prescription response is a response to a previously sent CancelRx. It may indicate whether the prescription was successfully canceled or already filled or the cancel denied.
- The prescriber system will always send a CancelRx and a NewRx, regardless of the type of change.
- MTMServiceRequest – A processor is requesting the provider provide a service to a patient.
- NewRxRequest – A request from a pharmacy to a prescriber for a new prescription for a patient.
- ClinicalInfoRequest - A request between healthcare providers for patient-centric clinical information.
- ClinicalInfoResponse – A response to the ClinicalInfoRequest to indicate if the information will/can be provided. The ClinicalInfoResponse can contain an attachment of the Clinical Information.
- PAInitiationResponse Transaction – A response from a payer to a prescriber with the information required to submit a prior authorization request for a specified patient and drug.
- PAResponse Transaction – A response from a payer to a prescriber with the status of a PARequest.
- PAAppealResponse Transaction – A response from a payer to a prescriber with the information required to submit a prior authorization appeal or the status of a specific prior authorization.
- PACancelResponse Transaction – A response from a payer to a prescriber about a cancellation.
- Verify - This verification is the response to a NewRx, RxChangeResponse, RefillResponse, or CancelRx or other transaction sent previously with a return receipt requested by the prescriber system. It indicates that the pharmacy or other entity has received the request.
- Error - Errors can be received under two different conditions. It is necessary to examine the recipient in the <To> field and <MessageID> field to determine its meaning. If the source is Mailbox and the trace number refers to the just sent GetMessage, then an error has been discovered by Mailbox in the GetMessage request itself. Otherwise, it refers to a previously sent transaction that was found to be in error by the pharmacy's or other entity's system.
- Status - A status in response to a GetMessage indicates that no more mail is available. Indirectly, then, any response to a GetMessage request other than a Status implies that additional mail *may* be in the mailbox and, therefore, solicits another GetMessage. A Status is never mailboxed and cannot contain an error.

Figure 2 illustrates a sample communication between a prescriber system and Mailbox where a RefillRequest is received as mail. Notice that the last transaction from Mailbox is a Status in response to a GetMessage. This means that no additional mail is available.

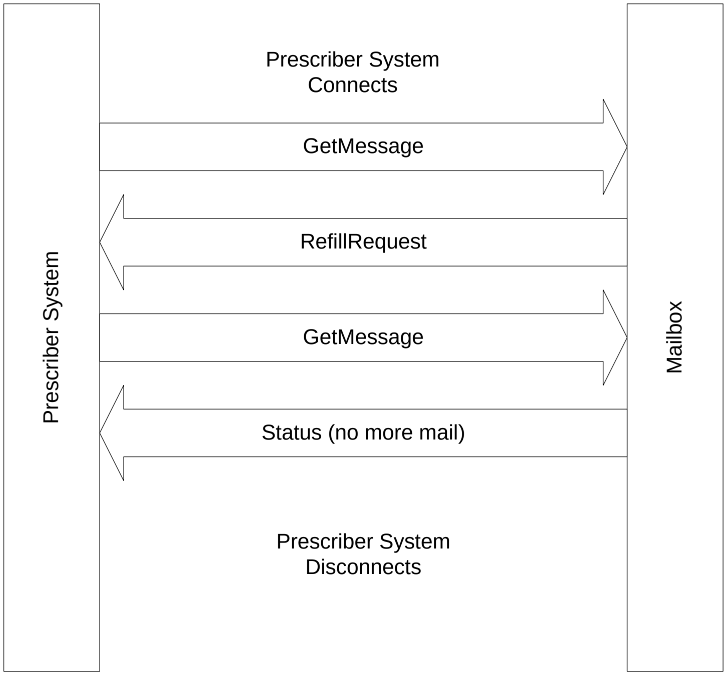


Figure 2. Sample exchange between a prescriber system and Mailbox in a dial scenario, where mail is delivered.

The RefillRequest is destined for, and therefore addressed to, a prescriber. The GetMessage, however, is addressed to the Mailbox.

In one type of Mailbox configuration, the prescriber sends the NewRx to the Mailbox. The Mailbox responds with a Status response. The pharmacy system dials in and asks for transactions (GetMessage). The Mailbox responds with the NewRx. The pharmacy asks for another transaction (GetMessage) and the Mailbox responds with a Status (no more mail). See **Figure 3**.

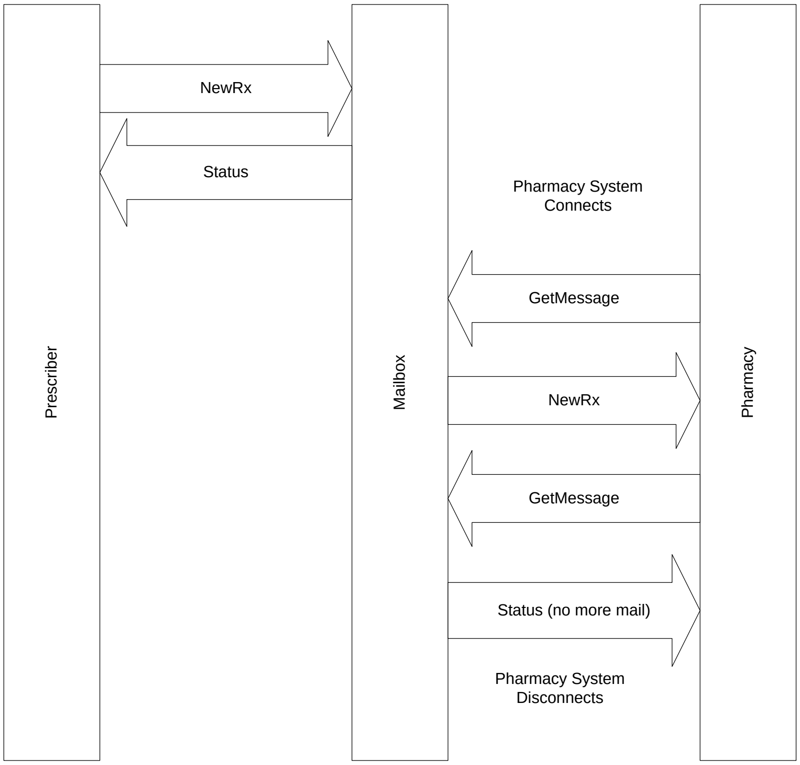


Figure 3. Flow for a successful NewRx Transaction in a Mailbox configuration. Prescriber system is directly connected to Mailbox; Pharmacy system is a dial connection to Mailbox.

In a Combination configuration where the prescriber system is dial and the pharmacy is on a dedicated line (for example), once the NewRx transaction (for example) arrives at the Mailbox it will be forwarded to the pharmacy system as soon as it is possible to do so. The pharmacy system is responsible for generating the Status response.

In dedicated line configurations, it is the responsibility of the pharmacy headquarters system to distribute the transaction to the appropriate pharmacy based on the recipient identification — the <To> field in the transaction. This field contains the NCPDP Provider ID Number of the pharmacy. This distribution should be as expeditious as possible, since the system becomes ineffective when the patient arrives at the pharmacy to pick up the prescription prior to the arrival of the transaction. NCPDP is not in a position to recommend the best strategy to achieve this effect, but delays should be avoided if possible.

The response transaction can be returned by either the pharmacy headquarters system or the pharmacy management system. **It cannot be responded to by both.** The pharmacy management system is the preferred responder, if this is consistent with the pharmacy architecture.

Figure 4 illustrates a sample communication between a pharmacy and the Mailbox where a RefillResponse from a previous RefillRequest and a NewRx is received as mail. The GetMessage is addressed to the Mailbox. The RefillResponse and the NewRx transactions are addressed to the pharmacy. Notice that the last transaction from the Mailbox is a Status in response to a GetMessage. The Status will contain appropriate codes (and text) to signify that no additional mail is available.

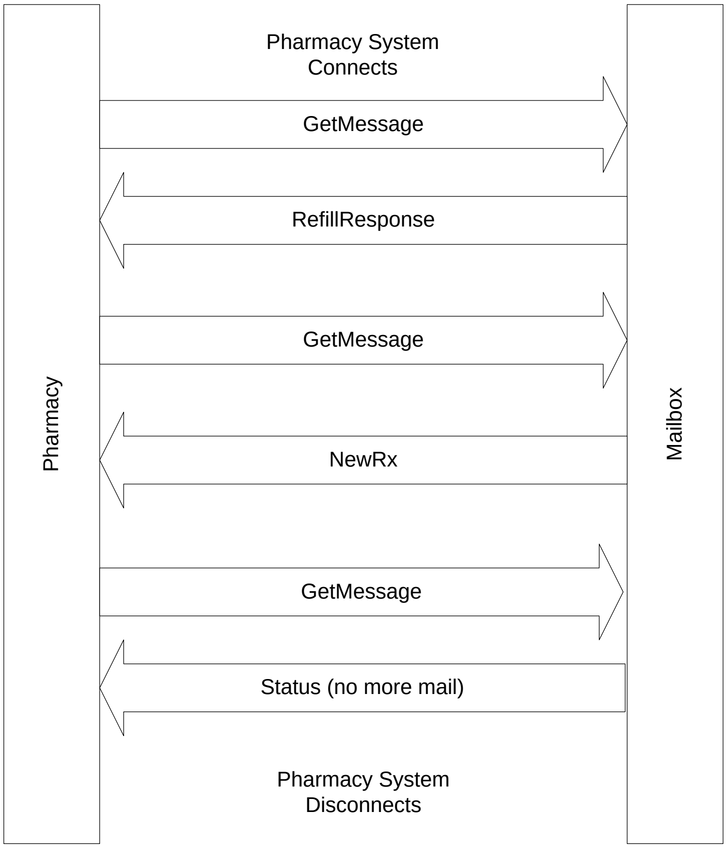


Figure 4. Sample exchange between a pharmacy system and the Mailbox where two pieces of mail are delivered.

It is important that in every connection the pharmacy or prescriber system attempts to obtain any mail the Mailbox may be holding, even if that is not the specific purpose of the connection.

In the scenario where a Mailbox dials the pharmacy or prescriber system to notify of mail, the Mailbox will establish a connection to the system, using any appropriate log on and sync up requirements. The prescriber or pharmacy system will then generate a GetMessage. The Mailbox will respond with the first piece of mail. The prescriber or pharmacy system initiates a second GetMessage and the Mailbox returns the next piece of mail (if there is one). This conversation continues until the Mailbox responds with a Status of No More Mail. The pharmacy or prescriber may send other transactions if they are ready to be sent to the Mailbox during the same conversation.

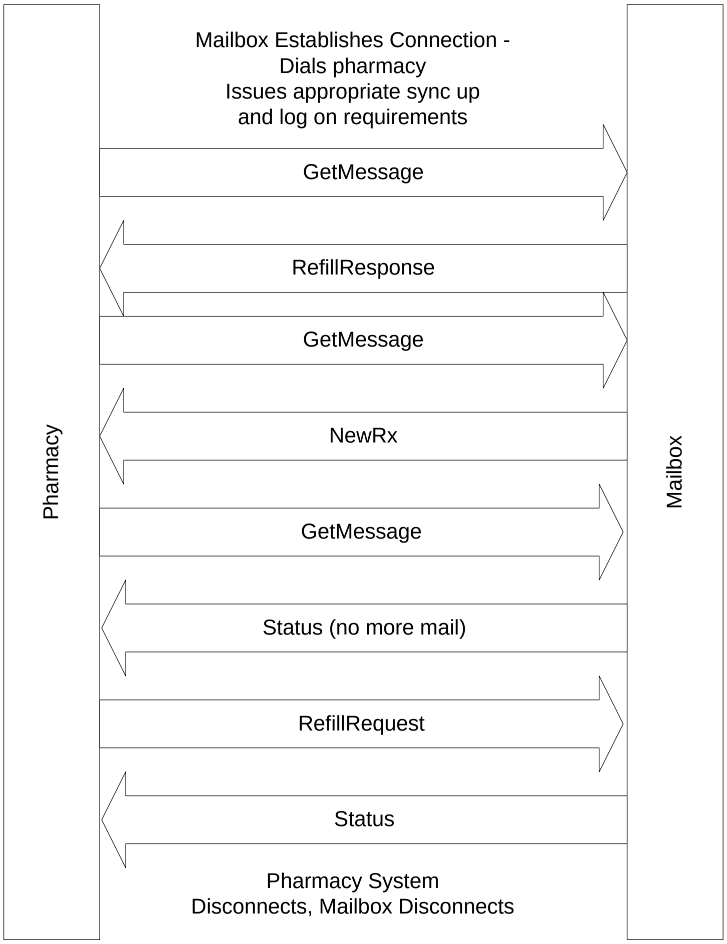

Pharmacy
Mailbox
Mailbox Establishes Connection - Dials pharmacy Issues appropriate sync up and log on requirements
GetMessage
RefillResponse
GetMessage
NewRx
GetMessage
Status (no more mail)
RefillRequest
Status
Pharmacy System Disconnects, Mailbox Disconnects

Figure 5. Scenario where the Mailbox dials out to the pharmacy when mail needs to be delivered. After mail has been received, the pharmacy system has a transaction to be sent before the connection is broken.

Note: The GetMessage transaction is not used for Central Fill functionality.

7.2 STATUS TRANSACTION

Status transactions always indicate that all is well with the acceptance of the request. **Status cannot be mailboxed and may not contain an error.** The Status cannot be mailboxed because it is necessary to recognize the response from a Mailbox as different than the mail actually to be delivered. The Status, as a standalone response, cannot be confused with mail that is to be delivered to the requester. In **Figure 2**, the Status is used to signify that all mail has been delivered. In **Figure 3**, the Status is used to denote successful acceptance of the NewRx by the prescriber system. Later in **Figure 3,** the status is used to signify that all mail has been delivered.

Therefore, a Status can be used to reply to the following:

- RefillRequest
- RefillResponse
- NewRx
- RxChangeRequest
- RxChangeResponse
- RxFill
- CancelRx
- CancelRxResponse
- GetMessage (no more mail)
- PasswordChange
- Verify
- Error
- RxHistoryRequest
- RxHistoryResponse
- Census
- Resupply
- MTMServiceRequest
- MTMServiceResponse
- NewRxRequest
- NewRxResponseDenied
- ClinicalInfoRequest
- ClinicalInfoResponse
- PAInitiationRequest
- PAInitiationResponse
- PARequest
- PAResponse
- PAAppealRequest
- PAAppealResponse
- PACancelRequest
- PACancelResponse
- CFInventoryList
- CFProductInquiry
- CFProductInquiryResponse
- CFRxOrderRequest
- CFRxOrderCompletion
- CFRxOrderCancel
- CFManifest

Figure 6 indicates the flow for such a transaction where the pharmacy management system has received the transaction on a direct connect.

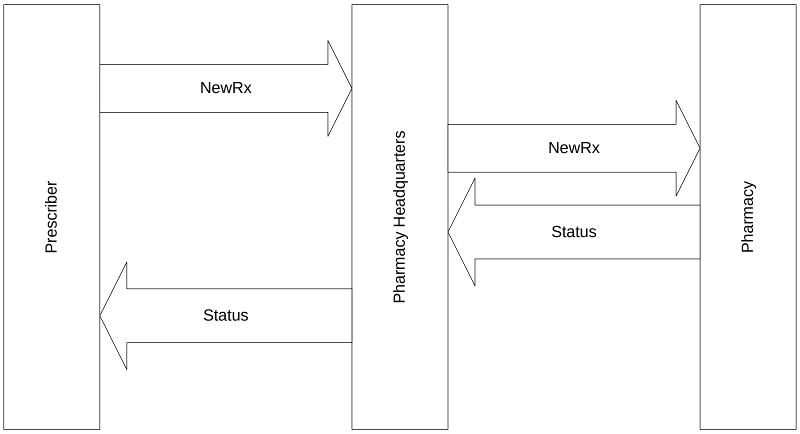


Figure 6. Flow for a successful NewRx Transaction on a direct connect.

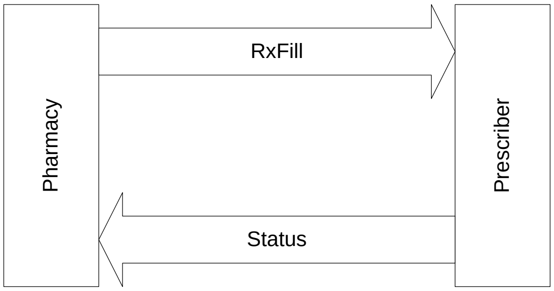


Figure 7. Flow for a successful RxFill Transaction (<FillStatus><Filled>) between a pharmacy and prescriber system on a direct connect.

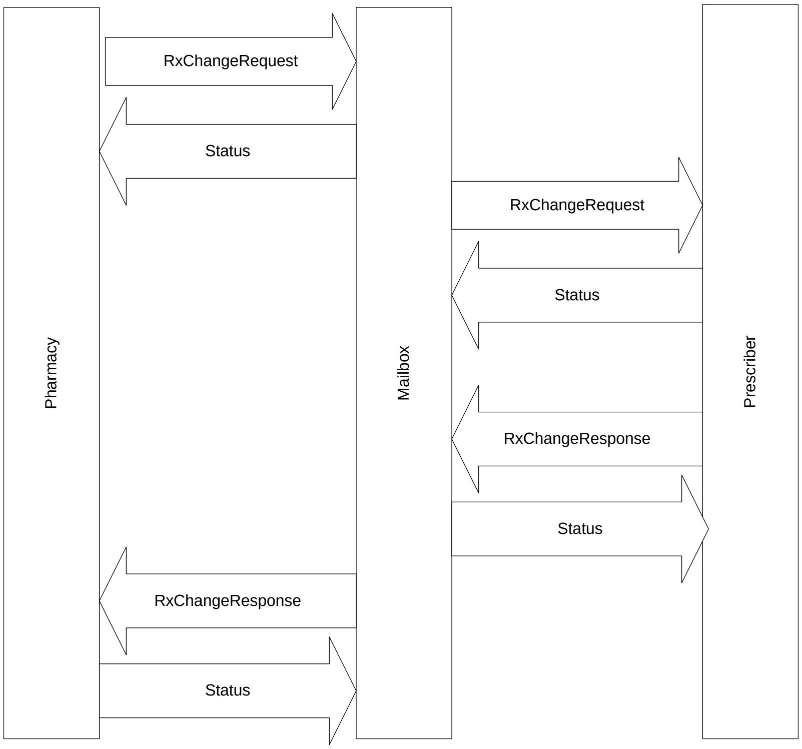


Figure 8. Flow for a successful RxChangeRequest and Response. Pharmacy and prescriber system might both be dedicated.

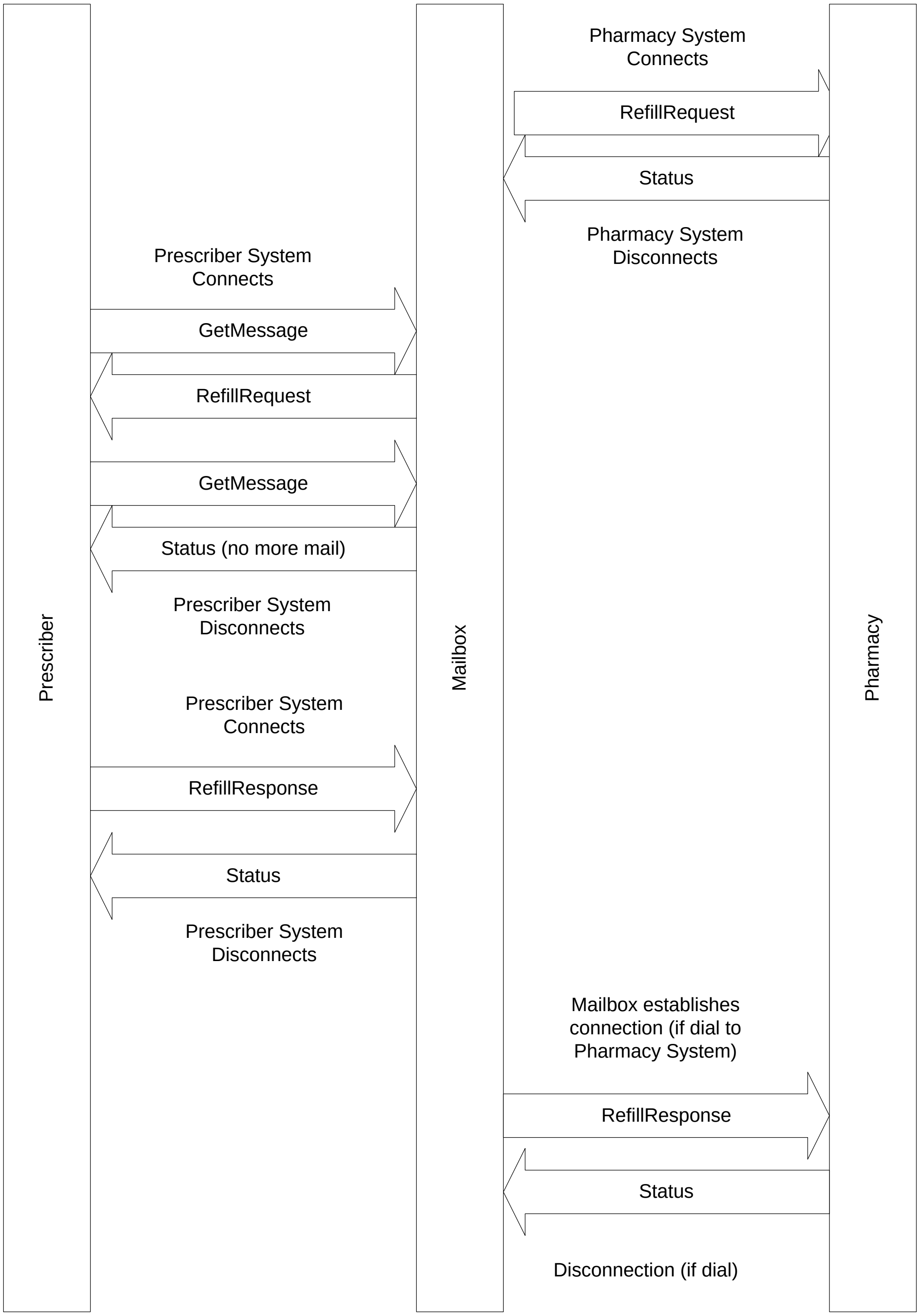

Prescriber
Mailbox
Pharmacy
Pharmacy System Connects
RefillRequest
Status
Pharmacy System Disconnects
Prescriber System Connects
GetMessage
RefillRequest
GetMessage
Status (no more mail)
Prescriber System Disconnects
Prescriber System Connects
RefillResponse
Status
Prescriber System Disconnects
Mailbox establishes connection (if dial to Pharmacy System)
RefillResponse
Status
Disconnection (if dial)

Figure 9. Flow for a RefillResponse. The prescriber system is dial to the Mailbox. The pharmacy is direct or able to receive dial from Mailbox.

7.3 ERROR TRANSACTION

Error transactions always indicate that there is a problem with the transaction. An Error can be mailboxed, as it may be signifying to the originator that a transaction was unable to be delivered or encountered problems in the acceptance. In **Figure 11**, the example shows how the Error may be mailboxed. Note that the Error must be a different response than a Status, since the communication between the prescriber and the Mailbox must clearly denote the actions taking place - the Error is a response being delivered on behalf of the pharmacy system, and the Status signifies no more mail.

Below, **Figure 10** illustrates the flow for a NewRx that is found to contain an error by the pharmacy headquarters system. **Figure 11** illustrates the flow for a NewRx that is found to contain an error by the pharmacy management system.

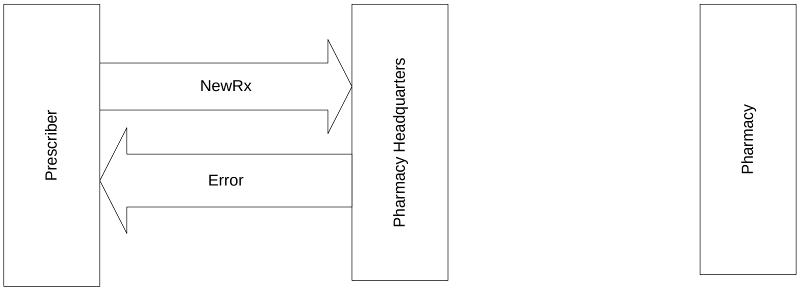


Figure 10. Flow for a NewRx Transaction where an error is detected by the Pharmacy Headquarters System.

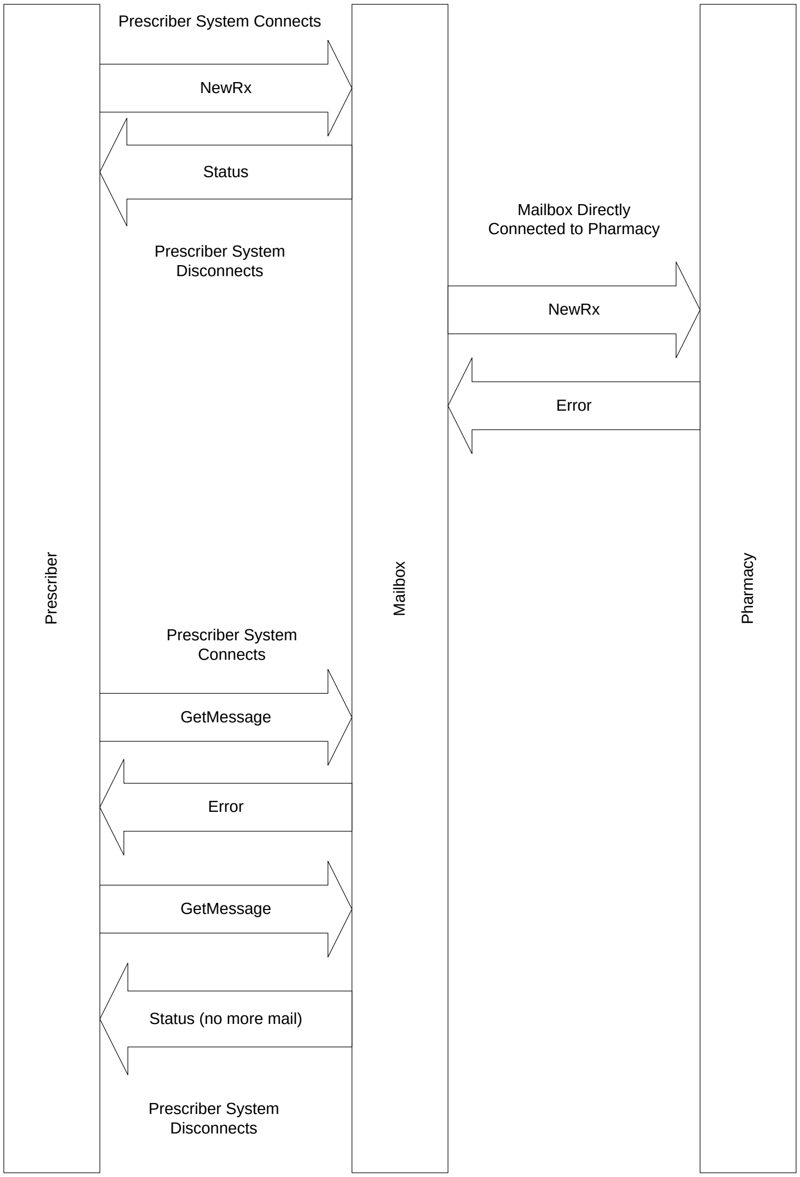

Prescriber System Connects
NewRx
Status
Prescriber System Disconnects
Mailbox Directly Connected to Pharmacy
NewRx
Error
Prescriber
Mailbox
Pharmacy
Prescriber System Connects
GetMessage
Error
GetMessage
Status (no more mail)
Prescriber System Disconnects

Figure 11. Flow for an Error response, in a Mailbox configuration. Prescriber system is dial and Pharmacy system is a direct connect to Mailbox.

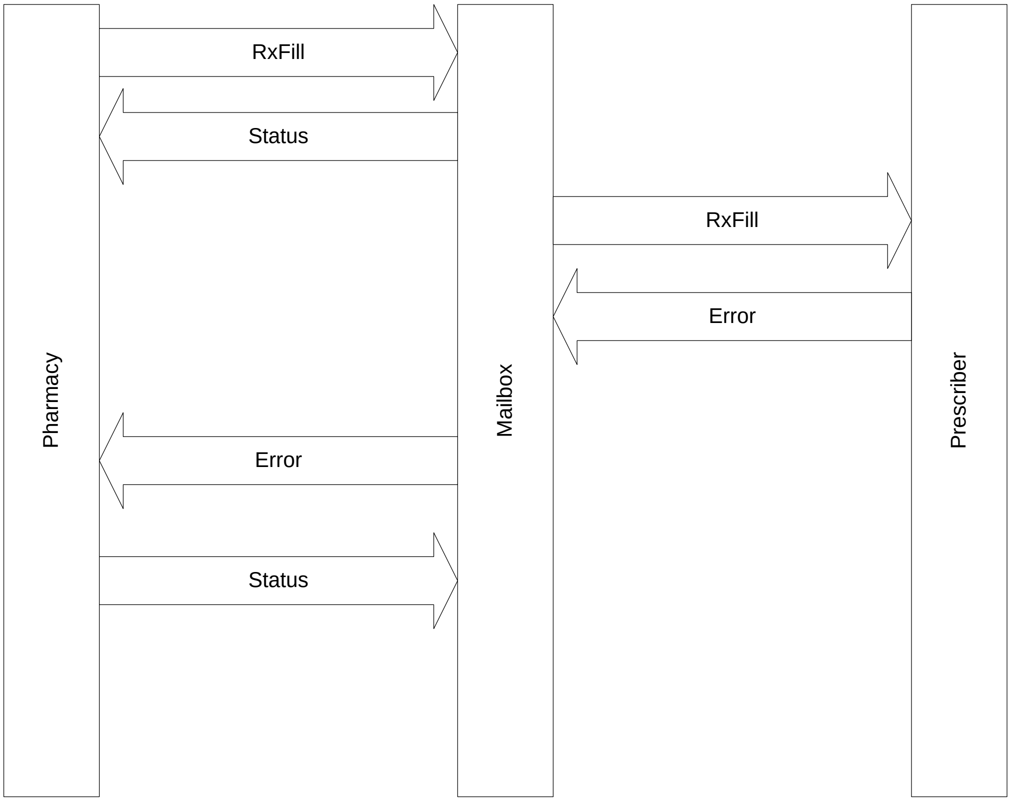


Figure 12. Flow for a RxFill Transaction (<FillStatus><Filled>) when the Prescriber Management System detects an error. The pharmacy and prescriber system are connected via direct connect to Mailbox.

7.4 VERIFY TRANSACTION

The Verify can be received as mail in response to a previous transactions requesting return receipt. The receiving system is responsible for generating such a transaction at an appropriate time. The appropriate time is at the discretion of the developer.

The purpose of the "return receipt" and Verify is to communicate to the sender that the receiver has received the transaction. Therefore, a Verify should be sent as close to the actual "receiving" of the transaction as possible. It is recommended that the Verify be sent when the transaction is received by the receiving system.

Three possible scenarios are shown below. (Not all transactions are listed)

- Upon receipt of a NewRx, RxChangeResponse, RefillResponse, or CancelRx transaction as mail and which requests a return receipt, the pharmacy system could in the same connection generate a Verify transaction.
- After having received a NewRx, RxChangeResponse, RefillResponse, CancelRx, or Resupply transaction, which requests a return receipt, but after the connection to the Mailbox has been terminated, a new call to the Mailbox could be initiated and the Verify transaction could be sent.
- After having received a NewRx, RxChangeResponse, RefillResponse, CancelRx, or Resupply transaction, which requests a receipt, but after the connection to the Mailbox has been terminated, the pharmacy system could generate a verification transaction, but not send it. The next time the pharmacy system calls the Mailbox, the Verify transaction can be delivered as part of the connection.

Figure 13 illustrates an example of the flow for the first case above.

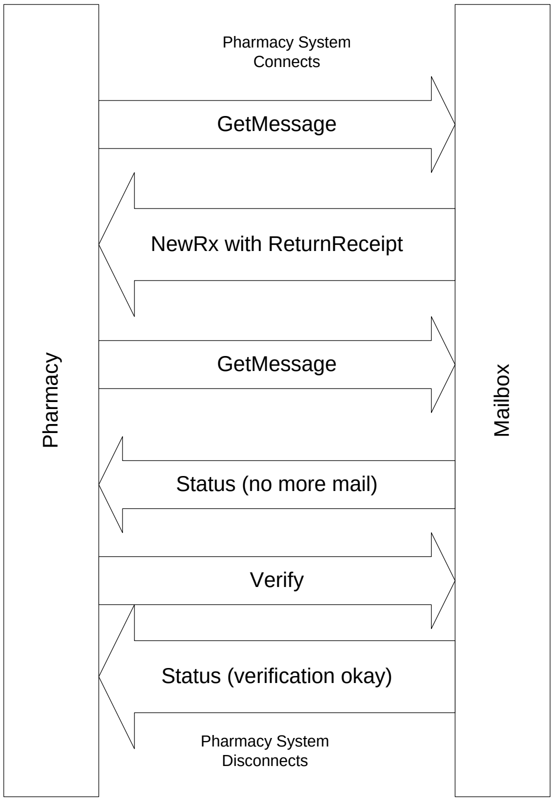


Figure 13. Sample Flow for a NewRx requesting a <ReturnReceipt> and the Corresponding Verify Transaction.

Figure 14 illustrates an example of the flow for the second case above.

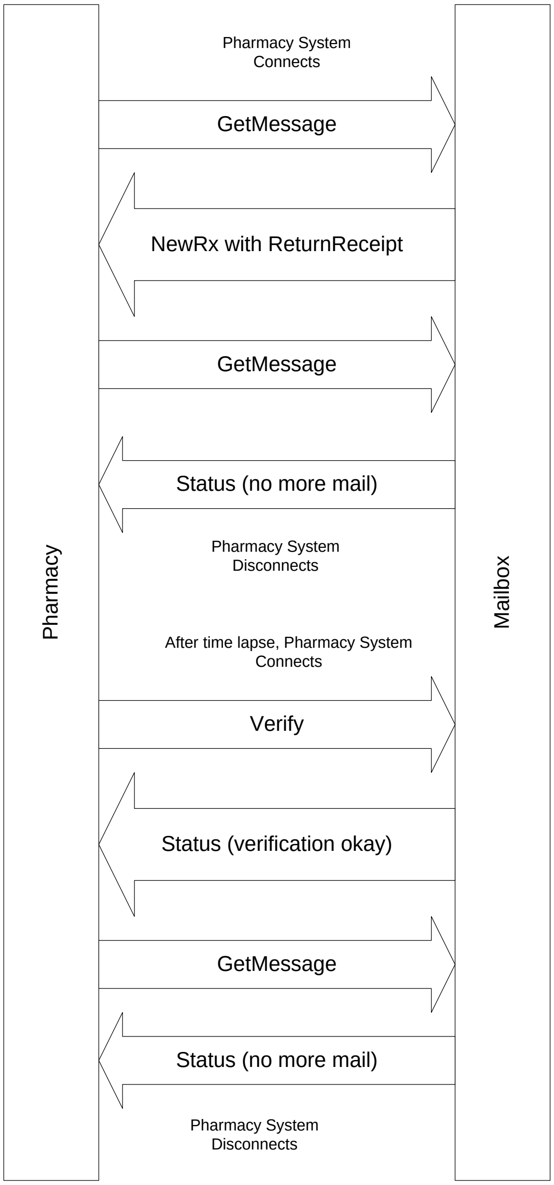


Figure 14. Sample Flow for a NewRx requesting a <ReturnReceipt> and the Corresponding Verify Transaction.

Notice that in **Figure 14** the pharmacy system issues a GetMessage after the Verify transaction is performed. It is important that in every connection the system attempts to obtain any mail the Mailbox may be holding, even if that is not the specific purpose of the connection.

Three other possible scenarios are shown below. (Note: not all transactions are listed)

- Upon receipt of an RxChangeRequest, RxFill, RefillRequest, or CancelRxResponse transaction as mail and which requests a return receipt, the prescriber system could in the same connection generate a verification transaction.
- After having received an RxChangeRequest, RxFill, RefillRequest, or CancelRxResponse transaction, which requests a return receipt, but after the connection to the Mailbox has been terminated, a new call to the Mailbox could be initiated and the verification transaction could be sent.
- After having received an RxChangeRequest, RxFill, RefillRequest, or CancelRxResponse

transaction, which requests a receipt, but after the connection to the Mailbox has been terminated, the prescriber system could generate a verification transaction, but not send it. When next the prescriber system calls the Mailbox, the verification transaction can be delivered as part of the connection.

Figures 15 and **16** illustrate the flows for a NewRx and a corresponding Verify transaction, where a <ReturnReceipt> is requested by the sender. Either of these two flows are acceptable. However, NCPDP recommends the flow illustrated in **Figure 15**, if this is consistent with the architecture of the pharmacy systems.

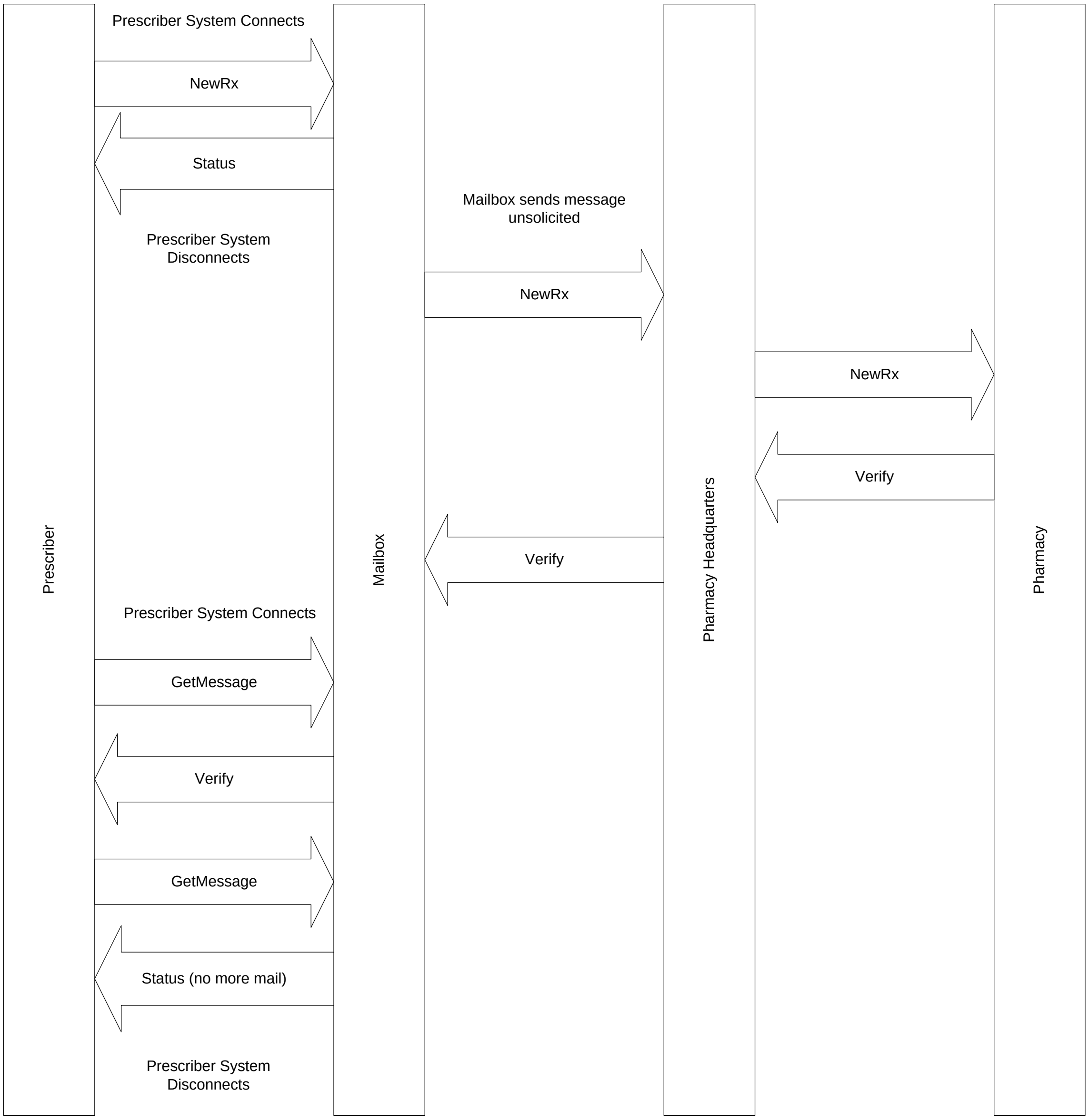

Prescriber
Mailbox
Pharmacy Headquarters
Pharmacy
Prescriber System Connects
NewRx
Status
Prescriber System Disconnects
Mailbox sends message unsolicited
NewRx
NewRx
Verify
Verify
Prescriber System Connects
GetMessage
Verify
GetMessage
Status (no more mail)
Prescriber System Disconnects

Figure 15. Flow for a NewRx Transaction Containing a <ReturnReceipt> requested. Verify is used as the response. Recommended Usage. Prescriber system is dial up; pharmacy system is direct connect to the Mailbox.

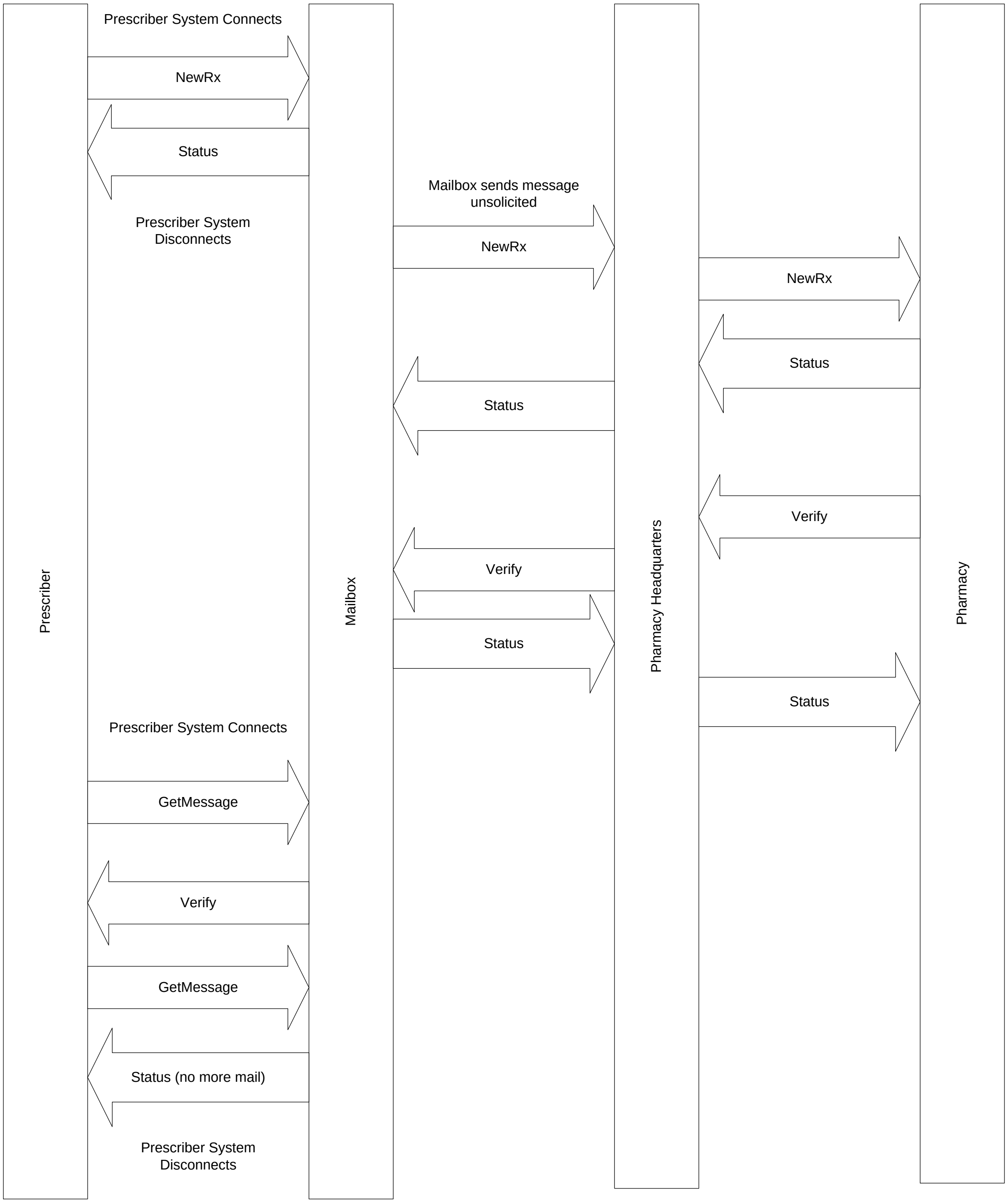
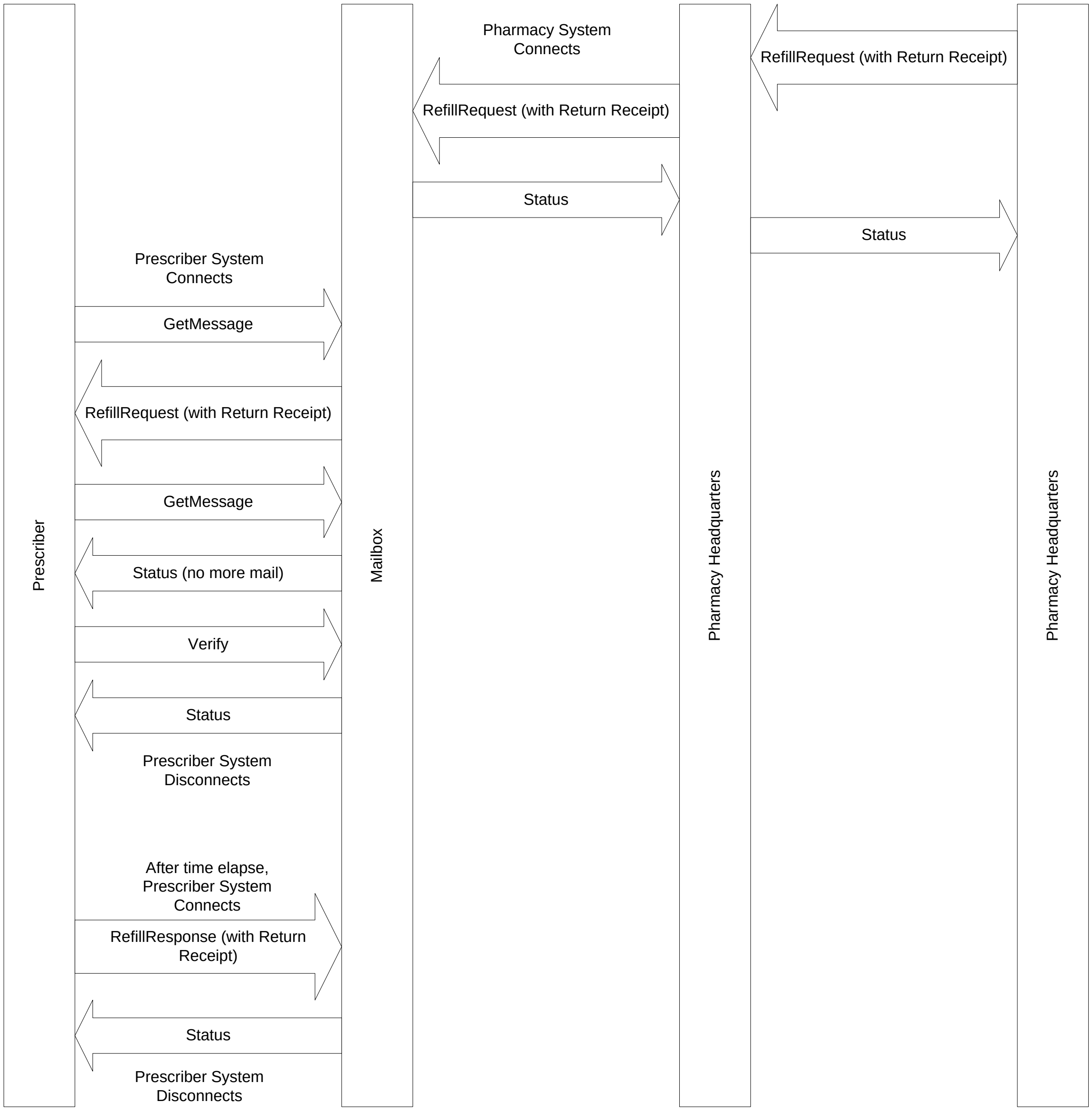


Figure 16. Flow for a NewRx Transaction Containing a <ReturnReceipt> requested. Verify is generated as a "free standing" transaction. Optional usage. Prescriber System is dial up; pharmacy system is direct connect to the Mailbox.

Figure 17 illustrates the complete flow for a RefillRequest transaction with a <ReturnReceipt> request and a RefillResponse also with a <ReturnReceipt> request. This is the most complex set of transactions supported by the system. As was noted above, the pharmacy headquarters may respond to a transaction containing <ReturnReceipt> requested directly with a Verify (recommended) or issue a Status and a "free standing" Verify transaction. **Figure 17** illustrates the latter case. Please see section "*Verify Transaction"* in the NCPDP ***XML Standard*** for more information.


Prescriber
Mailbox
Pharmacy Headquarters
Pharmacy Headquarters
RefillRequest (with Return Receipt)
Pharmacy System Connects
RefillRequest (with Return Receipt)
Status
Status
Prescriber System Connects
GetMessage
RefillRequest (with Return Receipt)
GetMessage
Status (no more mail)
Verify
Status
Prescriber System Disconnects
After time elapse, Prescriber System Connects
RefillResponse (with Return Receipt)
Status
Prescriber System Disconnects

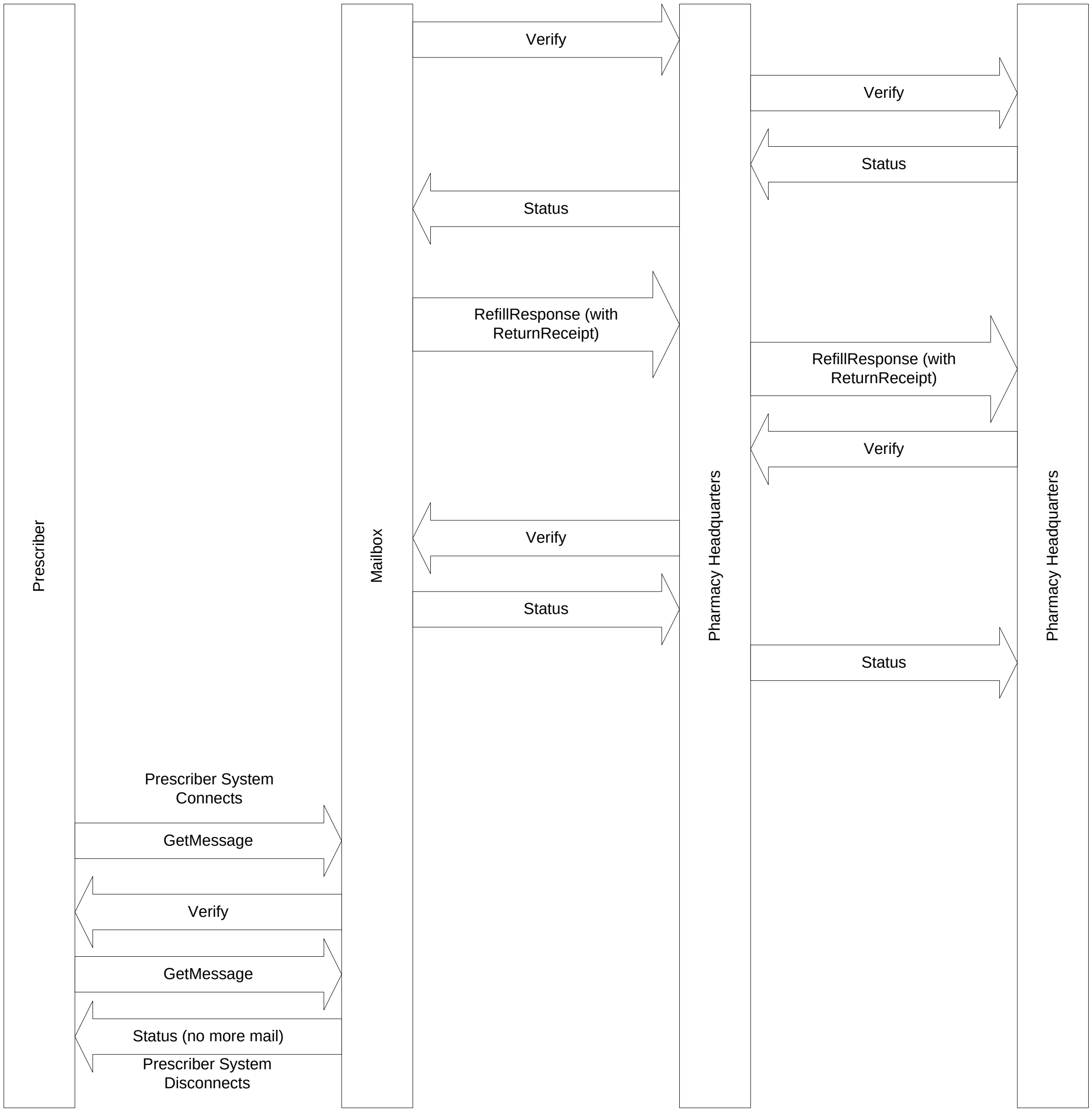


Figure 17. Flow for a Refill Function where both the Pharmacy and the Prescriber request <ReturnReceipt>. Prescriber system is dial, pharmacy system is direct connect.

These same models apply to any transaction exchange that supports <ReturnReceipt> and the Verify transaction.

7.5 PASSWORD CHANGE TRANSACTION

All transactions sent to the Mailbox may require passwords. The Mailbox will maintain passwords at the prescriber and pharmacy system level and therefore each prescriber and pharmacy system will have a unique password. This transaction serves the purpose of allowing the password for a pharmacy or prescriber system to be changed by providing the current password and a new password. Once this transaction has been performed, all future transactions must use the new password. At some future time, the prescriber or pharmacy system may choose to change their password. This transaction can once again be sent to update the Mailbox's file.

To provide an easy mechanism to get pharmacy or prescriber system started without the Mailbox keying in a password into a database (not recommended as this is a security risk), the first transaction performed by a pharmacy or prescriber system should be a PasswordChange with identical old and new passwords. This action will cause the Mailbox to establish this password as the password to be used from that point forward. This operation can only be performed once and will not work again and no other transactions will be successful until this operation is performed.

Figure 18 illustrates the flow associated with the PasswordChange. The "*Transmission Examples*" section of this document contains examples of an initial password establishing transaction and a normal change password transaction.

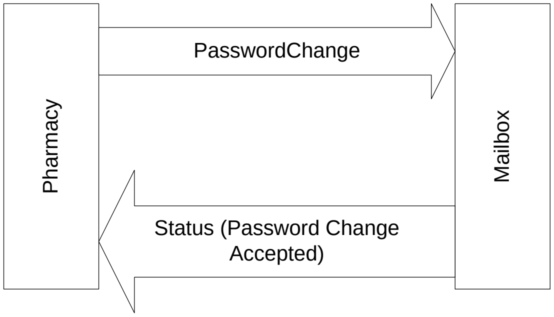


Figure 18. Flow for a PasswordChange Transaction

Like the GetMessage, this transaction is addressed to Mailbox in this scenario. Direct connection partners will need to determine rules for passwords, but the PasswordChange can be used as well.

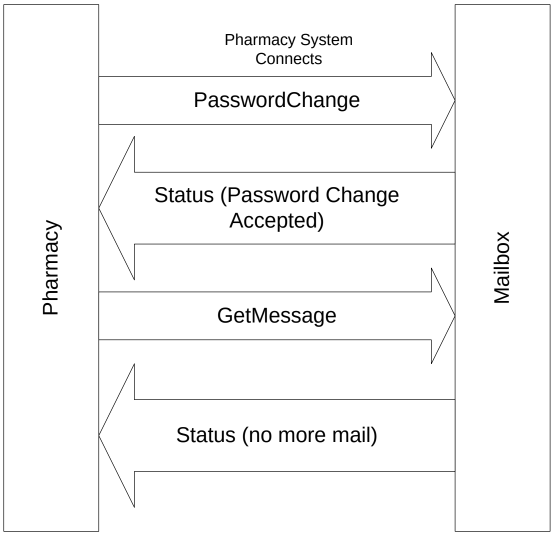


Figure 19. Flow for a PasswordChange Transaction where additional mail is also requested

In **Figure 19**, while the pharmacy or prescriber has sent in a PasswordChange, mail pickup can also take place in the same session. If the Mailbox has mail to deliver, the GetMessage dialogue will continue until the Mailbox responds with a Status of no more mail.

All requests and responses have two "tie back" fields. The first is <MessageID> which relates to the transmission. The second "tie back" field is <RelatesToMessageID> which is used to link transactions. See "*Trace Number Usage*" for usage rules.

8. STRUCTURE QUICK REFERENCE

Legend

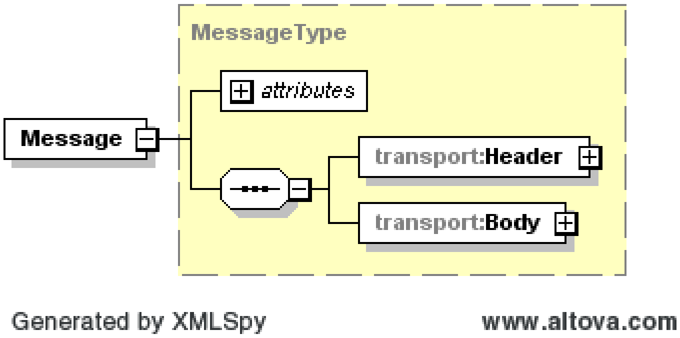
□	**Mandatory**
⬚	**Situational**
X	**Not used**

Note: Annotations in element diagrams provide instructional usage. Where a value may be cited (for example in element <Pharmacy>, the annotation may read “Value P2. When the recipient is a pharmacy, one loop is required for the pharmacy” – the value P2 refers to a value in the NCPDP ***External Code List***. While explicit tags do not need values, the annotation was included to provide guidance to lists in the NCPDP ***External Code List****.*

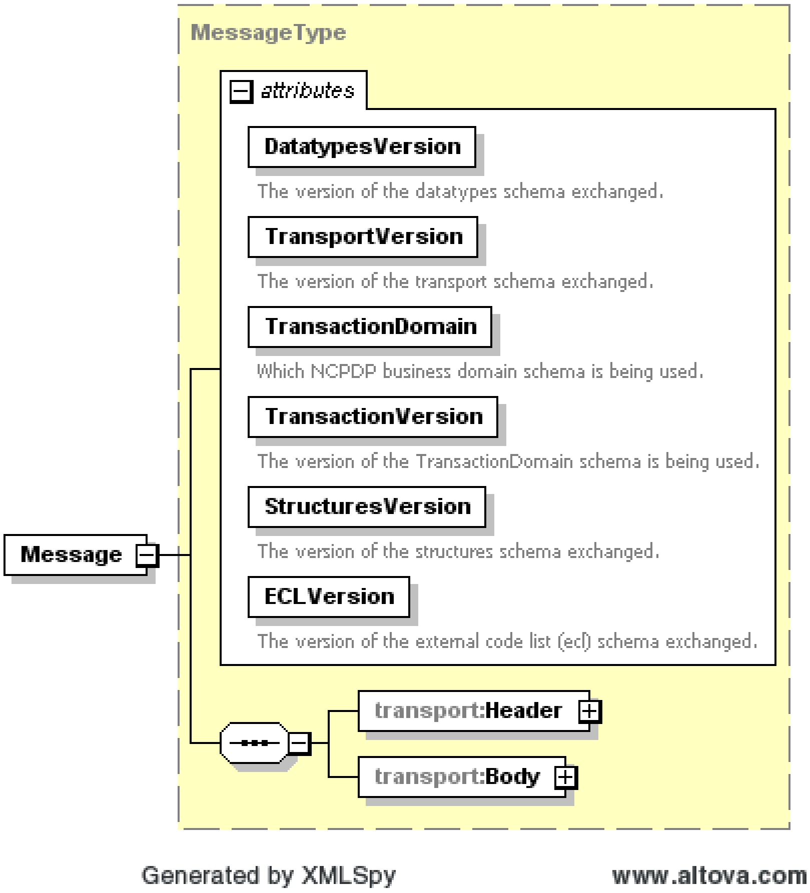
The model-driven environment supports the reuse of elements where possible. The transactions supported by this document provide the implementer with the ability to create a superset transaction of all applicable elements, reuse the superset, and only use the elements supported in the transaction subset. Elements in the superset that are not supported in the transaction subset are marked with the “X” Not used. Not used elements are not allowed to be sent.

8.1 TRANSACTION STRUCTURE

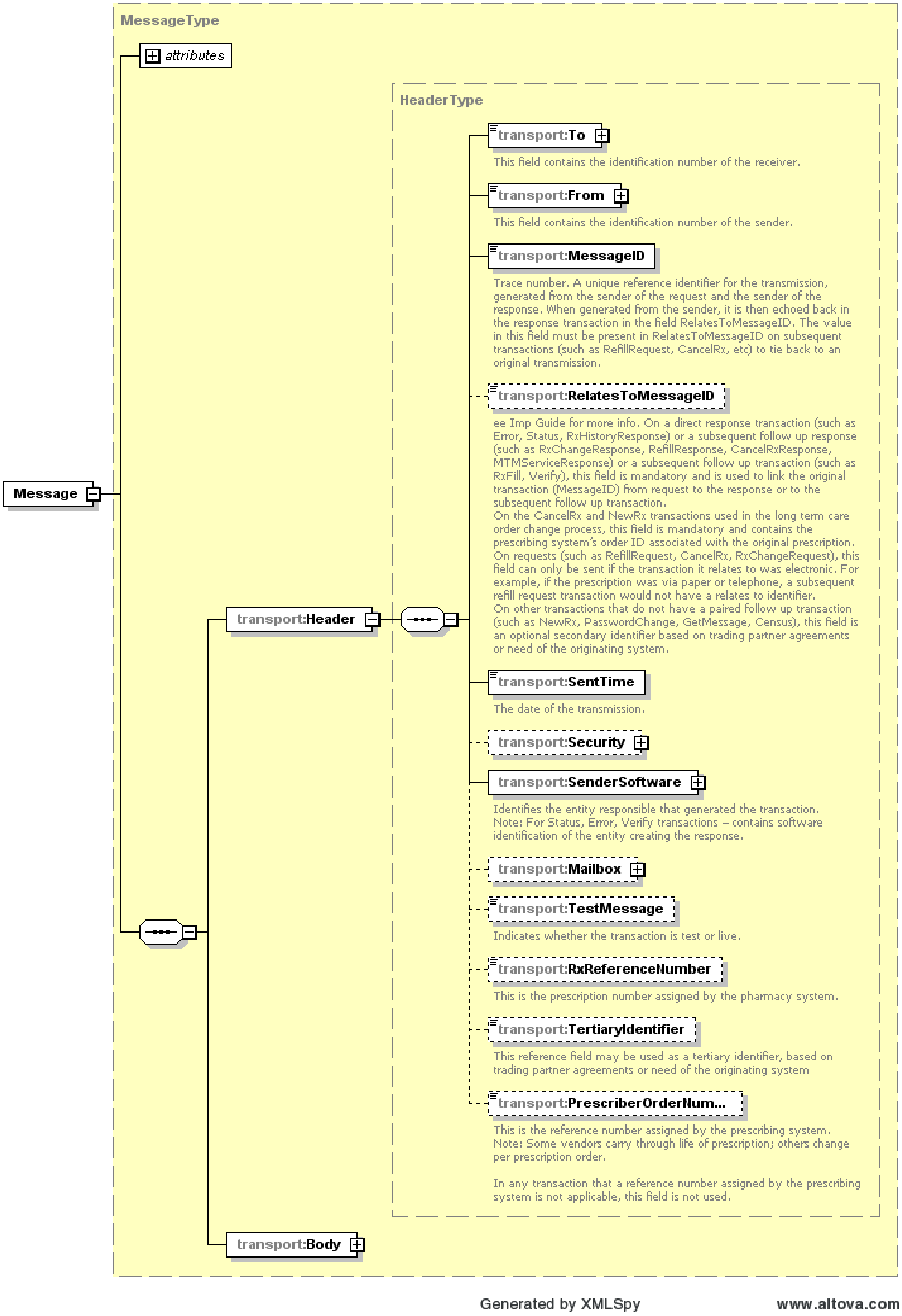
Each transaction of an XML transaction includes a <Header> and a <Body>.



The <Message> contains attributes that describe to the receiver which schemas the receiver has used to generate this transaction and the transaction version used.


MessageType
attributes
DatatypesVersion
The version of the datatypes schema exchanged.
TransportVersion
The version of the transport schema exchanged.
TransactionDomain
Which NCPDP business domain schema is being used.
TransactionVersion
The version of the TransactionDomain schema is being used.
StructuresVersion
The version of the structures schema exchanged.
ECLVersion
The version of the external code list (ecl) schema exchanged.
Message
transport:Header
transport:Body
Generated by XMLSpy
www.altova.com

The <Header> consists of important routing, trace, and identifier elements.



This field contains the identification number of the receiver.

This field contains the identification number of the sender.

Trace number. A unique reference identifier for the transmission, generated from the sender of the request and the sender of the response. When generated from the sender, it is then echoed back in the response transaction in the field RelatesToMessageID. The value in this field must be present in RelatesToMessageID on subsequent transactions (such as RefillRequest, CancelRx, etc) to tie back to an original transmission.

ee Imp Guide for more info. On a direct response transaction (such as Error, Status, RxHistoryResponse) or a subsequent follow up response (such as RxChangeResponse, RefillResponse, CancelRxResponse, MTMServiceResponse) or a subsequent follow up transaction (such as RxFill, Verify), this field is mandatory and is used to link the original transaction (MessageID) from request to the response or to the subsequent follow up transaction.
On the CancelRx and NewRx transactions used in the long term care order change process, this field is mandatory and contains the prescribing system's order ID associated with the original prescription.
On requests (such as RefillRequest, CancelRx, RxChangeRequest), this field can only be sent if the transaction it relates to was electronic. For example, if the prescription was via paper or telephone, a subsequent refill request transaction would not have a relates to identifier.
On other transactions that do not have a paired follow up transaction (such as NewRx, PasswordChange, GetMessage, Census), this field is an optional secondary identifier based on trading partner agreements or need of the originating system.

The date of the transmission.

Identifies the entity responsible that generated the transaction. Note: For Status, Error, Verify transactions – contains software identification of the entity creating the response.

Indicates whether the transaction is test or live.

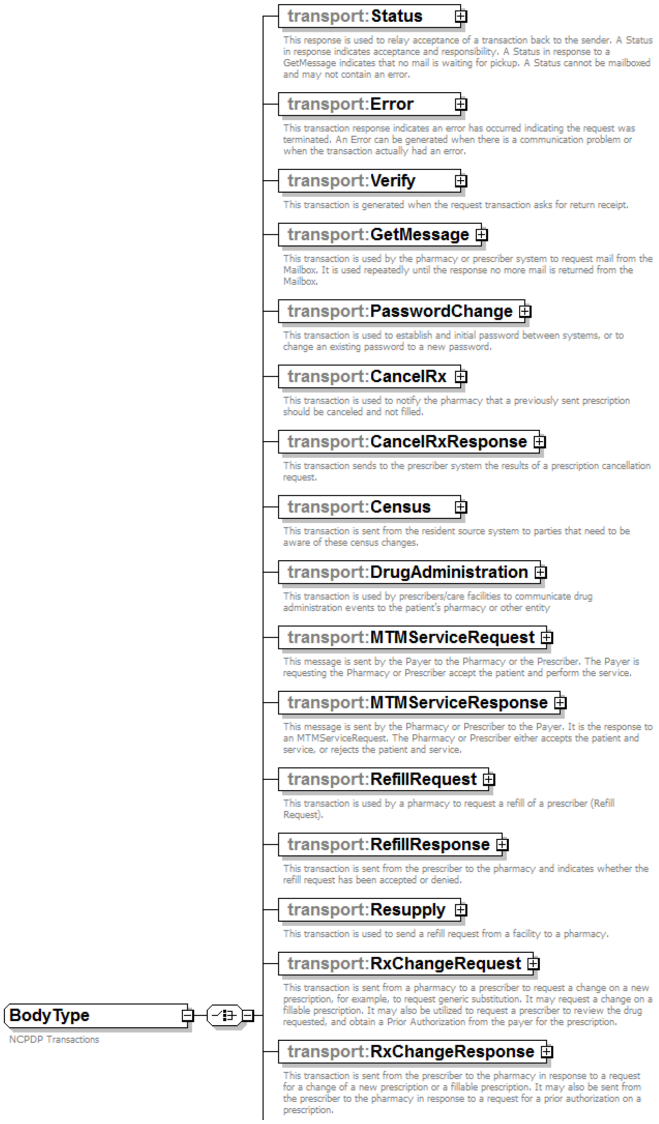
This is the prescription number assigned by the pharmacy system.

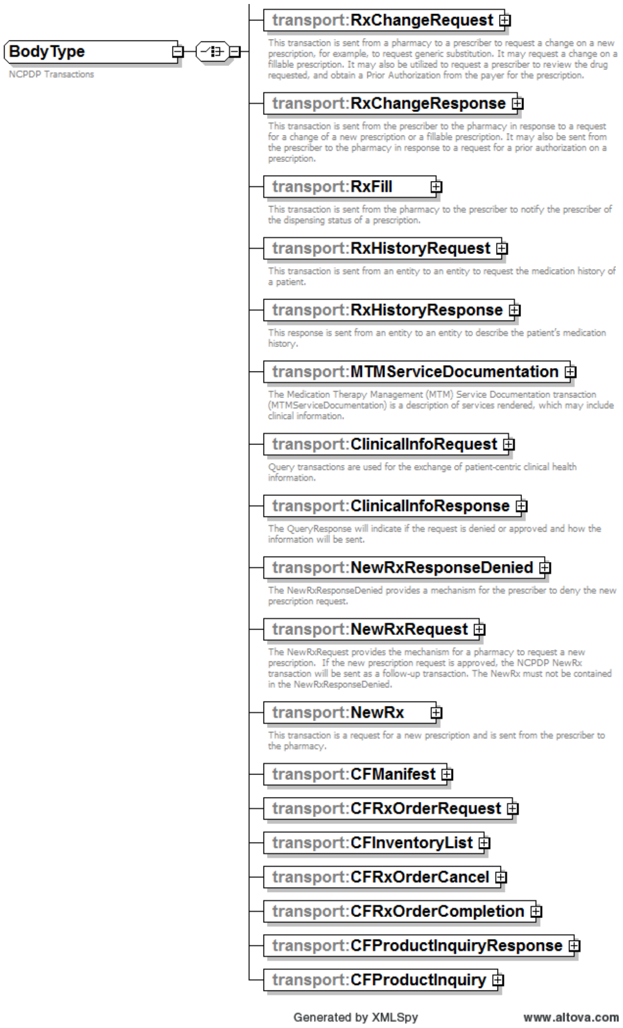
This reference field may be used as a tertiary identifier, based on trading partner agreements or need of the originating system

This is the reference number assigned by the prescribing system. Note: Some vendors carry through life of prescription; others change per prescription order.

In any transaction that a reference number assigned by the prescribing system is not applicable, this field is not used.

The <Body> consists of types of transactions.

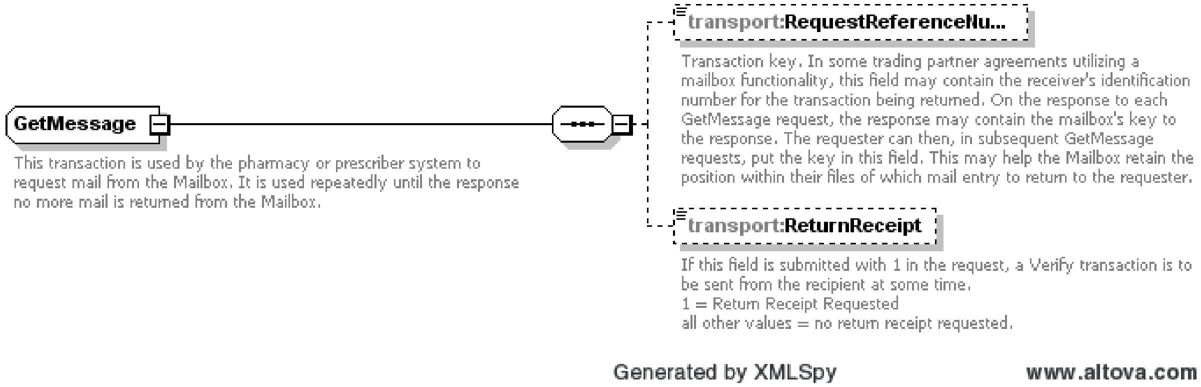

transport:Status
This response is used to relay acceptance of a transaction back to the sender. A Status in response indicates acceptance and responsibility. A Status in response to a GetMessage indicates that no mail is waiting for pickup. A Status cannot be mailboxed and may not contain an error.
transport:Error
This transaction response indicates an error has occurred indicating the request was terminated. An Error can be generated when there is a communication problem or when the transaction actually had an error.
transport:Verify
This transaction is generated when the request transaction asks for return receipt.
transport:GetMessage
This transaction is used by the pharmacy or prescriber system to request mail from the Mailbox. It is used repeatedly until the response no more mail is returned from the Mailbox.
transport:PasswordChange
This transaction is used to establish and initial password between systems, or to change an existing password to a new password.
transport:CancelRx
This transaction is used to notify the pharmacy that a previously sent prescription should be canceled and not filled.
transport:CancelRxResponse
This transaction sends to the prescriber system the results of a prescription cancellation request.
transport:Census
This transaction is sent from the resident source system to parties that need to be aware of these census changes.
transport:DrugAdministration
This transaction is used by prescribers/care facilities to communicate drug administration events to the patient's pharmacy or other entity
transport:MTMServiceRequest
This message is sent by the Payer to the Pharmacy or the Prescriber. The Payer is requesting the Pharmacy or Prescriber accept the patient and perform the service.
transport:MTMServiceResponse
This message is sent by the Pharmacy or Prescriber to the Payer. It is the response to an MTMServiceRequest. The Pharmacy or Prescriber either accepts the patient and service, or rejects the patient and service.
transport:RefillRequest
This transaction is used by a pharmacy to request a refill of a prescriber (Refill Request).
transport:RefillResponse
This transaction is sent from the prescriber to the pharmacy and indicates whether the refill request has been accepted or denied.
transport:Resupply
This transaction is used to send a refill request from a facility to a pharmacy.
transport:RxChangeRequest
This transaction is sent from a pharmacy to a prescriber to request a change on a new prescription, for example, to request generic substitution. It may request a change on a fillable prescription. It may also be utilized to request a prescriber to review the drug requested, and obtain a Prior Authorization from the payer for the prescription.
BodyType
NCPDP Transactions
transport:RxChangeResponse
This transaction is sent from the prescriber to the pharmacy in response to a request for a change of a new prescription or a fillable prescription. It may also be sent from the prescriber to the pharmacy in response to a request for a prior authorization on a prescription.


BodyType
NCPDP Transactions
transport:RxChangeRequest
This transaction is sent from a pharmacy to a prescriber to request a change on a new prescription, for example, to request generic substitution. It may request a change on a fillable prescription. It may also be utilized to request a prescriber to review the drug requested, and obtain a Prior Authorization from the payer for the prescription.
transport:RxChangeResponse
This transaction is sent from the prescriber to the pharmacy in response to a request for a change of a new prescription or a fillable prescription. It may also be sent from the prescriber to the pharmacy in response to a request for a prior authorization on a prescription.
transport:RxFill
This transaction is sent from the pharmacy to the prescriber to notify the prescriber of the dispensing status of a prescription.
transport:RxHistoryRequest
This transaction is sent from an entity to an entity to request the medication history of a patient.
transport:RxHistoryResponse
This response is sent from an entity to an entity to describe the patient's medication history.
transport:MTMServiceDocumentation
The Medication Therapy Management (MTM) Service Documentation transaction (MTMServiceDocumentation) is a description of services rendered, which may include clinical information.
transport:ClinicalInfoRequest
Query transactions are used for the exchange of patient-centric clinical health information.
transport:ClinicalInfoResponse
The QueryResponse will indicate if the request is denied or approved and how the information will be sent.
transport:NewRxResponseDenied
The NewRxResponseDenied provides a mechanism for the prescriber to deny the new prescription request.
transport:NewRxRequest
The NewRxRequest provides the mechanism for a pharmacy to request a new prescription. If the new prescription request is approved, the NCPDP NewRx transaction will be sent as a follow-up transaction. The NewRx must not be contained in the NewRxResponseDenied.
transport:NewRx
This transaction is a request for a new prescription and is sent from the prescriber to the pharmacy.
transport:CFManifest
transport:CFRxOrderRequest
transport:CFInventoryList
transport:CFRxOrderCancel
transport:CFRxOrderCompletion
transport:CFProductInquiryResponse
transport:CFProductInquiry
Generated by XMLSpy
www.altova.com

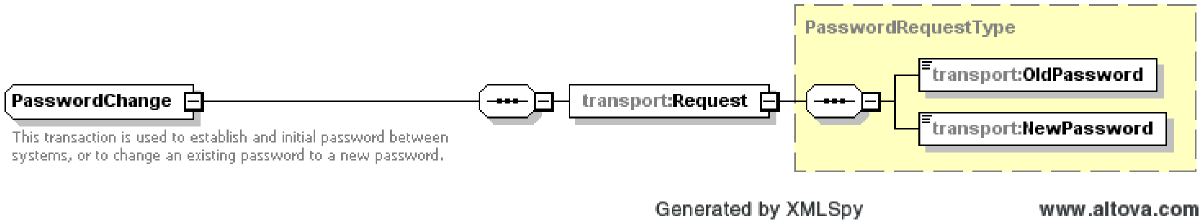
8.2 ELEMENT USAGE IN EACH TRANSACTION

Transactions consist of elements in groupings according to topic.

8.2.1 GETMESSAGE

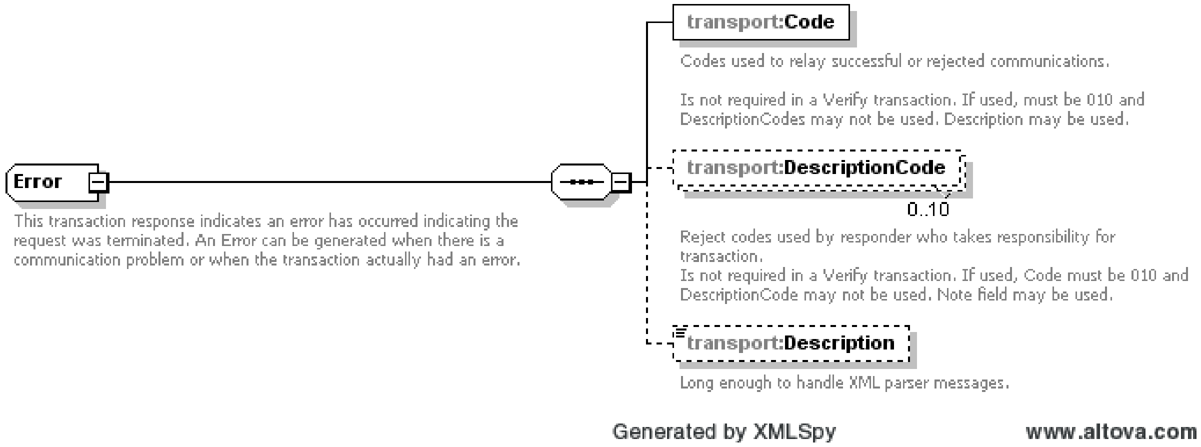

GetMessage
This transaction is used by the pharmacy or prescriber system to request mail from the Mailbox. It is used repeatedly until the response no more mail is returned from the Mailbox.
transport:RequestReferenceNu...
Transaction key. In some trading partner agreements utilizing a mailbox functionality, this field may contain the receiver's identification number for the transaction being returned. On the response to each GetMessage request, the response may contain the mailbox's key to the response. The requester can then, in subsequent GetMessage requests, put the key in this field. This may help the Mailbox retain the position within their files of which mail entry to return to the requester.
transport:ReturnReceipt
If this field is submitted with 1 in the request, a Verify transaction is to be sent from the recipient at some time.
1 = Return Receipt Requested
all other values = no return receipt requested.
Generated by XMLSpy
www.altova.com

8.2.2 PASSWORDCHANGE



This transaction is used to establish and initial password between systems, or to change an existing password to a new password.

8.2.3 ERROR



This transaction response indicates an error has occurred indicating the request was terminated. An Error can be generated when there is a communication problem or when the transaction actually had an error.

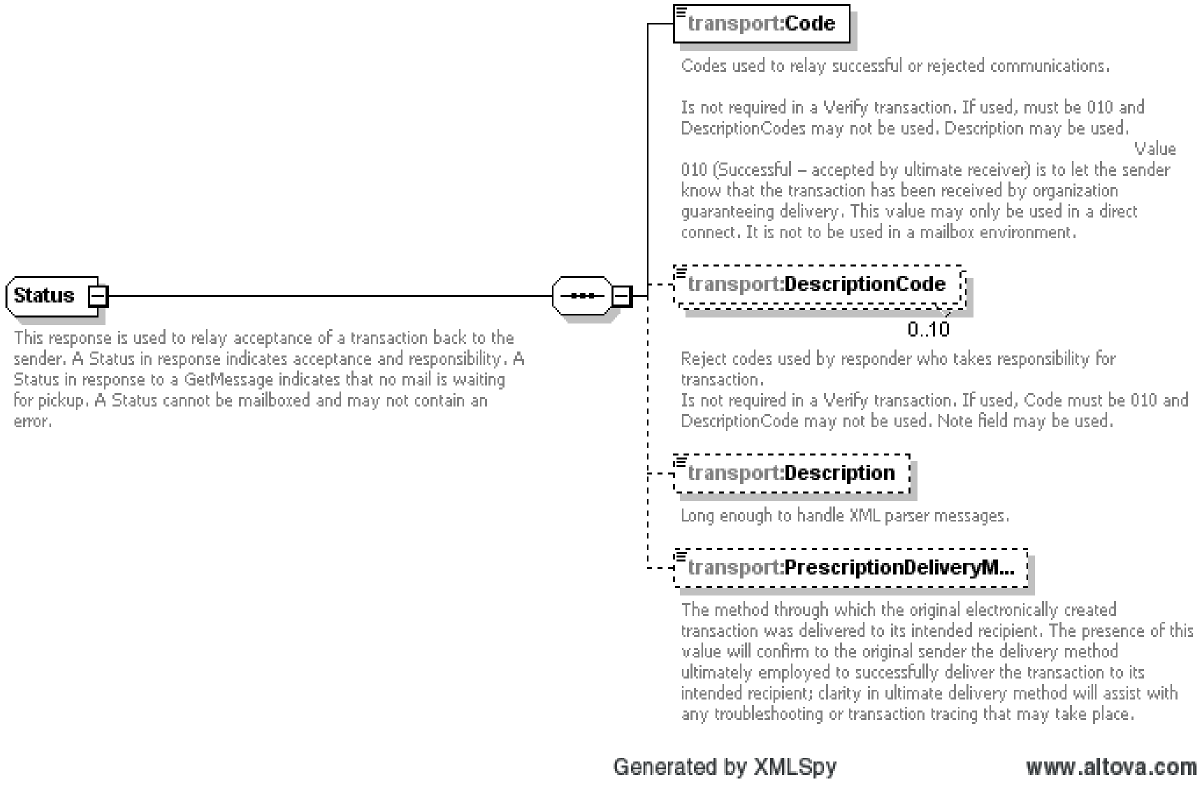
Codes used to relay successful or rejected communications.

Is not required in a Verify transaction. If used, must be 010 and DescriptionCodes may not be used. Description may be used.

Reject codes used by responder who takes responsibility for transaction.
Is not required in a Verify transaction. If used, Code must be 010 and DescriptionCode may not be used. Note field may be used.

Long enough to handle XML parser messages.

8.2.4 STATUS



This response is used to relay acceptance of a transaction back to the sender. A Status in response indicates acceptance and responsibility. A Status in response to a GetMessage indicates that no mail is waiting for pickup. A Status cannot be mailboxed and may not contain an error.

Codes used to relay successful or rejected communications.

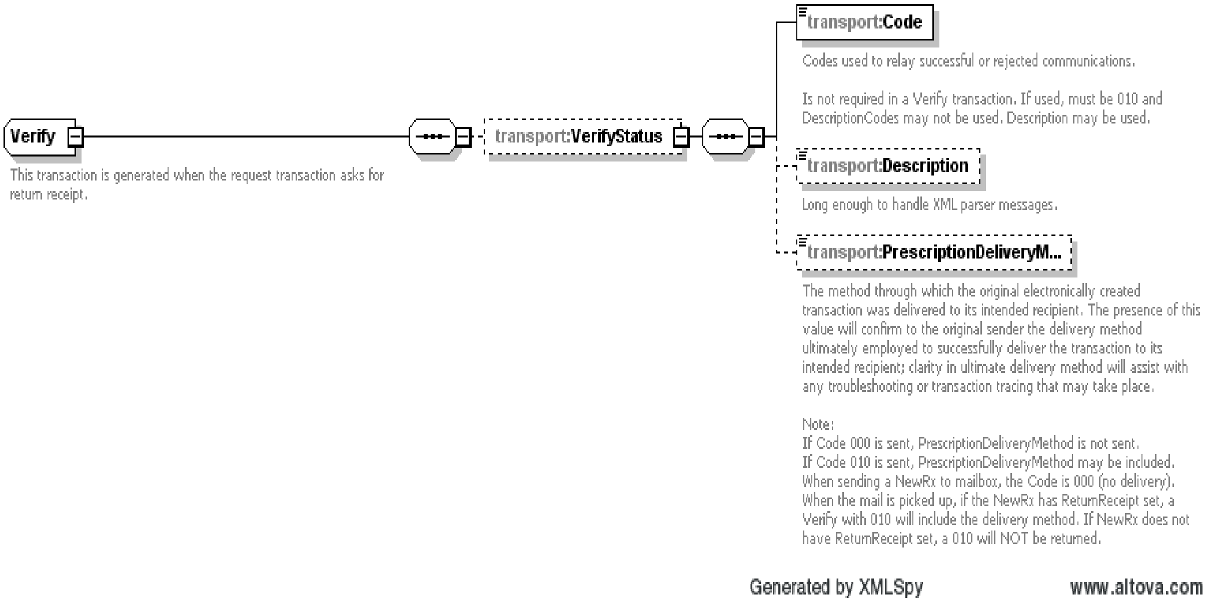
Is not required in a Verify transaction. If used, must be 010 and DescriptionCodes may not be used. Description may be used.
Value
010 (Successful – accepted by ultimate receiver) is to let the sender know that the transaction has been received by organization guaranteeing delivery. This value may only be used in a direct connect. It is not to be used in a mailbox environment.

Reject codes used by responder who takes responsibility for transaction.
Is not required in a Verify transaction. If used, Code must be 010 and DescriptionCode may not be used. Note field may be used.

Long enough to handle XML parser messages.

The method through which the original electronically created transaction was delivered to its intended recipient. The presence of this value will confirm to the original sender the delivery method ultimately employed to successfully deliver the transaction to its intended recipient; clarity in ultimate delivery method will assist with any troubleshooting or transaction tracing that may take place.

8.2.5 VERIFY



This transaction is generated when the request transaction asks for return receipt.

Codes used to relay successful or rejected communications.

Is not required in a Verify transaction. If used, must be 010 and DescriptionCodes may not be used. Description may be used.

Long enough to handle XML parser messages.

The method through which the original electronically created transaction was delivered to its intended recipient. The presence of this value will confirm to the original sender the delivery method ultimately employed to successfully deliver the transaction to its intended recipient; clarity in ultimate delivery method will assist with any troubleshooting or transaction tracing that may take place.

Note:
If Code 000 is sent, PrescriptionDeliveryMethod is not sent.
If Code 010 is sent, PrescriptionDeliveryMethod may be included.
When sending a NewRx to mailbox, the Code is 000 (no delivery).
When the mail is picked up, if the NewRx has ReturnReceipt set, a Verify with 010 will include the delivery method. If NewRx does not have ReturnReceipt set, a 010 will NOT be returned.

9. SPECIFIC ELEMENT DISCUSSION

This section discusses specific elements used within the XML format and the usage intended for this implementation. The schemas contain annotations to help the implementer with rules of usage.

9.1 MESSAGE ATTRIBUTES

The attributes of the Message contain the versions of the schemas used. The appropriate combinations of schemas for a specific publication are found in the NCPDP ***Basic Guide to Standards*** document. The version of the schemas must match exactly to the publication version shown in the NCPDP ***Basic Guide to Standards***.

The <TransactionDomain> must contain a valid value from the schema list. Transactions that are part of the NCPDP ***SCRIPT Standard Implementation Guide*** must contain the value "SCRIPT". Transactions that are part of the NCPDP ***Specialized Standard Implementation Guide*** must contain the value "Specialized". The transactions defined within this document (GetMessage, PasswordChange, Error, Verify, Status) must contain the value for the domain the transaction is being used in (SCRIPT or Specialized).

> For example, if you are in an electronic prescribing environment, the value "SCRIPT" is to be sent in <TransactionDomain> for Error, Status, etc.
>
> If using the specialized transactions, the value "Specialized" is to be sent in <TransactionDomain>.
>
> It is possible to be in a mixed environment, where it is appropriate to receive a Status that responds to a Census (Specialized) and a Status that responds to a NewRx (SCRIPT).

9.2 MAILBOX NOTE

When the Mailbox has mail to deliver (for example, NewRx, RxChangeRequest and RxChangeResponse, RefillRequest and RefillResponse, etc.), the Mailbox does not echo the trace numbers from the GetMessage <MessageID> and <RelatesToMessageID>, but rather sends the original data from the transaction, except for the removal of the password.

When the Mailbox has no mail to deliver, the Status will follow <MessageID> and <RelatesToMessageID> rules.

When the Mailbox is responding to a transaction for delivery (responding with a Status to a NewRx, for example), the Status will follow <MessageID> and <RelatesToMessageID> rules.

<RequestReferenceNumber> - In some trading partner agreements utilizing a mailbox functionality, this element may contain the receiver's identification number for the transaction being returned. On the response to each GetMessage, the response may contain the mailbox's key to the response. The requester can then, in subsequent GetMessage requests, put the key in this field. This may help the Mailbox retain the position within their files of which mail entry to return to the requester.

See section "*Transmission Examples*" for examples containing Mailbox functionality.

9.3 TRACE NUMBER USAGE

The <MessageID> and <RelatesToMessageID> rules are to be followed for assigning, responding, and referencing transaction activity except in the "*Mailbox Note*" above. Further, the *"Transmission Examples"* section should be reviewed for implementation and more detail. The following is summary guidelines of the usage.

Field Name	Remarks
<MessageID>	Unique reference identifier for the transmission, generated from the sender of the request and the sender of the response.
<RelatesToMessageID>	Specific requirements for long term care. Is used for linkage to response or to subsequent transactions.
<RxReferenceNumber>	The prescription number assigned by the pharmacy system.
<PrescriberOrderNumber>	The reference number assigned by the prescribing system.

Note: In NewRx transactions that begin with the prescriber (e.g. an "unsolicited" NewRx - a prescriber initiating a NewRx to a pharmacy not based on a response to a request from the pharmacy), the <RelatesToMessageID> is not sent as it is not applicable.

In NewRx transactions that begin with the pharmacy (e.g. a "solicited" NewRx - a pharmacy sending a NewRxRequest), the NewRx transaction **must** contain a <RelatesToMessageID>. The flow:

1. A pharmacy sends a NewRxRequest.
2. The prescriber approves the request and sends a NewRx.
3. The NewRx <RelatesToMessageID> **must** contain the <MessageID> from the NewRxRequest.

Note: If a NewRx is received with <RelatesToMessageID> containing an ID the pharmacy does not recognize, the pharmacy may choose to treat this as an "unsolicited" NewRx as it is a new prescription and may be treated as such. (However, the errant <RelatesToMessageID> is confusing and should be corrected.)

In the examples that follow trace and reference numbers are shown to follow through in examples. No intelligence is implied in the values used, or in any sequence.

9.3.1 EXAMPLE 1

Prescriber sends the original order, cancels the order, and then sends a new order.

	NewRx from Prescriber	CancelRx from Prescriber	NewRx from Prescriber
Field Name	Value		
<MessageID>	1234567	1234569	1234568
<RelatesToMessageID>		1234567	
<RxReferenceNumber>			
<PrescriberOrderNumber>	110088	110088	110081

	NewRx from Prescriber	**CancelRx from Prescriber**	**NewRx from Prescriber**
Field Name	**Value**		
	Status from Pharmacy	**Status from Pharmacy**	**Status from Pharmacy**
<MessageID>	ABC111	A66	A69
<RelatesToMessageID>	1234567	1234569	1234568
<RxReferenceNumber>			
<PrescriberOrderNumber>			

9.3.2 EXAMPLE 2

Prescriber sends a new prescription. Pharmacy reports two fill transactions.

	NewRx from Prescriber	**RxFill (partial fill) from Pharmacy**	**RxFill (partial fill) from Pharmacy**
Field Name	**Value**		
<MessageID>	1234567	3311	3433
<RelatesToMessageID>		1234567	1234567
<RxReferenceNumber>		PH456	PH456
<PrescriberOrderNumber>	110088	110088	110088
	Status from Pharmacy	**Status from Prescriber**	**Status from Prescriber**
<MessageID>	ABC11	8899	9988
<RelatesToMessageID>	1234567	3311	3433
<RxReferenceNumber>			
<PrescriberOrderNumber>			

9.3.3 EXAMPLE 3

Prescriber sends a new prescription. After the prescription runs out but is still active, the Pharmacy requests a refill authorization. Prescriber responds.

	NewRx from Prescriber	**RefillRequest from Pharmacy**	**RefillResponse from Prescriber**
Field Name	**Value**		
<MessageID>	2222	A22	2290
<RelatesToMessageID>		2222	A22
<RxReferenceNumber>		PH111	PH111
<PrescriberOrderNumber>	4444444	4444444	4444444
	Status from Pharmacy	**Status from Prescriber**	**Status from Pharmacy**
<MessageID>	A99	2288	A45

	NewRx from Prescriber	**RefillRequest from Pharmacy**	**RefillResponse from Prescriber**
Field Name	**Value**		
<RelatesToMessageID>	2222	A22	2290
<RxReferenceNumber>			
<PrescriberOrderNumber>			

9.3.4 EXAMPLE 4

Prescriber requests medication history.

	RxHistoryRequest from Prescriber
Field Name	**Value**
<MessageID>	991
<RelatesToMessageID>	
<RxReferenceNumber>	
<PrescriberOrderNumber>	
	RxHistoryResponse from Aggregator
<MessageID>	Z22
<RelatesToMessageID>	991
<RxReferenceNumber>	
<PrescriberOrderNumber>	

9.3.5 EXAMPLE 5

Prescriber sends a new prescription to a pharmacy via the pharmacy's mailbox entity. Pharmacy later sends fill status transactions to the prescriber.

	NewRx from Prescriber	**GetMessage from Pharmacy**	**GetMessage from Pharmacy**	**RxFill from Pharmacy**	**GetMessage from Prescriber**	**GetMessage from Prescriber**
Field Name	**Value**					
<MessageID>	1234567	A22	A25	A87	123562	123563
<RelatesToMessageID>				1234567		
<RxReferenceNumber>				PH456		
<PrescriberOrderNumber>	110088			110088		
	Status from Mailbox	**NewRx from Mailbox**	**Status (from Mailbox (no more mail)**	**Status from Mailbox**	**RxFill from Mailbox**	**Status from Mailbox (no more mail)**
<MessageID>	66666681	1234567	66666684	2223	A87	8898

	NewRx from Prescriber	GetMessage from Pharmacy	GetMessage from Pharmacy	RxFill from Pharmacy	GetMessage from Prescriber	GetMessage from Prescriber
Field Name	**Value**					
<RelatesToMessageID>	1234567		A25	A87	1234567	123563
<RxReferenceNumber>					PH456	
<PrescriberOrderNumber>		110088			110088	

9.3.6 Example 6

Pharmacy requesting a refill authorization, sending 2 follow-up requests. Prescriber responds to first request.

	NewRx from Prescriber	RefillRequest from Pharmacy	RefillRequest from Pharmacy Follow-up 1	RefillRequest from Pharmacy Follow-up 2	RefillResponse from Prescriber
Field Name	**Value**				
<MessageID>	2222	A22	B22	C22	2290
<RelatesToMessageID>		2222	A22	A22	A22 (or could be B22 or C22, depends on which request the prescriber is responding to)
<RxReferenceNumber>			PH111	PH111	PH111
<PrescriberOrderNumber>	4444444	4444444	4444444	4444444	4444444
<FollowUpRequest>			1	2	
	Status from Pharmacy	**Status from Prescriber**	**Status from Prescriber**	**Status from Prescriber**	**Status from Pharmacy**
<MessageID>	A99	2288	3399	4400	B99
<RelatesToMessageID>	2222	A22	B22	C22	2290
<RxReferenceNumber>					
<PrescriberOrderNumber>					

9.3.7 EXAMPLE 7

Pharmacy requests a NewRx from the Prescriber via the NewRxRequest transaction. Prescriber sends NewRx. The NewRx <RelatesToMessageID> must contain the <MessageID> from the NewRxRequest.

	NewRxRequest from Pharmacy	**NewRx from Prescriber**
Field Name	**Value**	
<MessageID>	1234567	1234569
<RelatesToMessageID>		1234567
<RxReferenceNumber>		
<PrescriberOrderNumber>		110088
	Status from Prescriber	**Status from Pharmacy**
<MessageID>	ABC111	A66
<RelatesToMessageID>	1234567	1234569
<RxReferenceNumber>		
<PrescriberOrderNumber>		

9.3.8 EXAMPLE 8

Pharmacy requests clinical info from the Prescriber via a ClinicalInfoRequest transaction.
The Prescriber responds initially, and then sends subsequent information.

Prescriber sends a ClinicalInfoResponse (Approved with a MessageToFollow). The <RelatesToMessageID> of any ClinicalInfoResponse transaction(s) must contain the <MessageID> from the ClinicalInfoRequest.

In this example, the subsequent ClinicalInfoResponse transactions contain the <RelatesToMessageID> from the ClinicalInfoRequest as the mandatory tieback. The <MessageToFollowMessageIdentifier> is sent in the <MessageID> of the subsequent ClinicalInfoResponse as a secondary tieback.

The <AdditionalMessageIndicator> is set to true on the first ClinicalInfoResponse, and to false on the second (and final) ClinicalInfoResponse transaction.

	ClinicalInfoRequest from Pharmacy	**ClinicalInfoResponse from Prescriber**	**ClinicalInfoResponse from Prescriber**
Field Name	**Value**		
<MessageID>	1234567	1234569	4567890
<RelatesToMessageID>		1234567	1234567
<AdditionalMessageIndicator>		true	false
<MessageToFollowMessageIdentifier>		4567890	
<RxReferenceNumber>			
<PrescriberOrderNumber>			
	Status from Prescriber	**Status from Pharmacy**	**Status from Pharmacy**
<MessageID>	ABC111	A66	A77
<RelatesToMessageID>	1234567	1234569	4567890
<RxReferenceNumber>			
<PrescriberOrderNumber>			

9.3.9 EXAMPLE 9

The same example as 8 except the prescriber has multiple ClinicalInfoResponses to send to fulfill the request.
The Prescriber responds initially, and then sends subsequent information in multiple transactions.

Pharmacy requests clinical info from the Prescriber via a ClinicalInfoRequest transaction.
Prescriber sends a ClinicalInfoResponse (Approved with a MessageToFollow). The <RelatesToMessageID> of any ClinicalInfoResponse transaction must contain the <MessageID> from the ClinicalInfoRequest.

In this example, each ClinicalInfoResponse contains the <RelatesToMessageID> from the ClinicalInfoRequest as the mandatory tieback. The <MessageToFollowMessageIdentifier> is sent in the <MessageID> of the subsequent ClinicalInfoResponse as a secondary tieback.

The <AdditionalMessageIndicator> is set to true on the first ClinicalInfoResponse, and to false on the second (and final) ClinicalInfoResponse transaction.

	ClinicalInfoRequest from Pharmacy	**ClinicalInfoResponse from Prescriber**	**ClinicalInfoResponse from Prescriber**	**ClinicalInfoResponse from Prescriber**
Field Name	**Value**			
<MessageID>	1234567	1234569	4567890	55555
<RelatesToMessageID>		1234567	1234567	1234567
<AdditionalMessageIndicator>		true	true	false
<MessageToFollowMessageIdentifier>		4567890	55555	
<RxReferenceNumber>				
<PrescriberOrderNumber>				
	Status from Prescriber	**Status from Pharmacy**	**Status from Pharmacy**	**Status from Pharmacy**
<MessageID>	ABC111	A66	A77	A80
<RelatesToMessageID>	1234567	1234569	4567890	55555
<RxReferenceNumber>				
<PrescriberOrderNumber>				

9.3.10 EXAMPLE 10

The same example as 8 except the prescriber has multiple ClinicalInfoResponses to send to fulfill the request.
The Prescriber responds initially, and then sends subsequent information in multiple transactions.

Pharmacy requests clinical info from the Prescriber via a ClinicalInfoRequest transaction.
Prescriber sends a ClinicalInfoResponse (Approved with a MessageToFollow). The <RelatesToMessageID> of any ClinicalInfoResponse transaction must contain the <MessageID> from the ClinicalInfoRequest.

In this example, each ClinicalInfoResponse contains the <RelatesToMessageID> from the ClinicalInfoRequest as the mandatory tieback. The <MessageToFollowMessageIdentifier> is able to be sent on the first ClinicalInfoResponse transaction. It is then sent in the <MessageID> of the second ClinicalInfoResponse as a secondary tieback, but in this example, the prescriber is unable to assign a <MessageToFollowMessageIdentifier> on the second ClinicalInfoResponse transaction. Therefore in the third ClinicalInfoResponse, the <MessageID> contains whatever identifier is assigned at that time.

The <AdditionalMessageIndicator> is set to true on the first and second ClinicalInfoResponse, and to false on the third (and final) ClinicalInfoResponse transaction.

	ClinicalInfoRequest from Pharmacy	ClinicalInfoResponse from Prescriber	ClinicalInfoResponse from Prescriber	ClinicalInfoResponse from Prescriber
Field Name	**Value**			
<MessageID>	1234567	1234569	4567890	666666
<RelatesToMessageID>		1234567	1234567	1234567
<AdditionalMessageIndicator>		true	true	false
<MessageToFollowMessageIdentifier>		4567890		
<RxReferenceNumber>				
<PrescriberOrderNumber>				
	Status from Prescriber	**Status from Pharmacy**	**Status from Pharmacy**	**Status from Pharmacy**
<MessageID>	ABC111	A66	A77	A80
<RelatesToMessageID>	1234567	1234569	4567890	666666
<RxReferenceNumber>				
<PrescriberOrderNumber>				

9.3.11 EXAMPLE 11

The same example as 8 except the prescriber has multiple ClinicalInfoResponses to send to fulfill the request.
The Prescriber responds initially, and then sends subsequent information in multiple transactions.

Pharmacy requests clinical info from the Prescriber via a ClinicalInfoRequest transaction.
Prescriber sends a ClinicalInfoResponse (Approved with a MessageToFollow). The <RelatesToMessageID> of any ClinicalInfoResponse transaction must contain the <MessageID> from the ClinicalInfoRequest.

In this example, each ClinicalInfoResponse contains the <RelatesToMessageID> from the ClinicalInfoRequest as the mandatory tieback. The <MessageToFollowMessageIdentifier> is not sent on the ClinicalInfoResponse transaction. Therefore in the subsequent ClinicalInfoResponse transactions, the <MessageID> contains whatever identifier is assigned at that time.

The <AdditionalMessageIndicator> is set to true on the first and second ClinicalInfoResponse, and to false on the third (and final) ClinicalInfoResponse transaction.

	ClinicalInfoRequest from Pharmacy	**ClinicalInfoResponse from Prescriber**	**ClinicalInfoResponse from Prescriber**	**ClinicalInfoResponse from Prescriber**
Field Name	**Value**			
<MessageID>	1234567	1234569	444444	666666
<RelatesToMessageID>		1234567	1234567	1234567
<AdditionalMessageIndicator>		true	true	false
<MessageToFollowMessageIdentifier>				
<RxReferenceNumber>				
<PrescriberOrderNumber>				
	Status from Prescriber	**Status from Pharmacy**	**Status from Pharmacy**	**Status from Pharmacy**
<MessageID>	ABC111	A66	A77	A80
<RelatesToMessageID>	1234567	1234569	444444	666666
<RxReferenceNumber>				
<PrescriberOrderNumber>				

9.3.12 EXAMPLE 12

This is an alternate to Example 8 where the Prescriber is able to respond immediately with the clinical information (the initial response to the ClinicalInfoRequest is a ClinicalInfoResponse).

Pharmacy requests clinical info from the Prescriber via a ClinicalInfoRequest transaction.
The Prescriber responds initially, and then provides subsequent information.

Prescriber sends a ClinicalInfoResponse (Approved with a MessageToFollow). The <RelatesToMessageID> of any ClinicalInfoResponse transaction must contain the <MessageID> from the ClinicalInfoRequest.

If the <MessageToFollowMessageIdentifier> is sent in the initial ClinicalInfoResponse, then subsequent ClinicalInfoResponse transactions must contain the value in <MessageID>.

	ClinicalInfoRequest from Pharmacy	**ClinicalInfoResponse from Prescriber**
Field Name	**Value**	
<MessageID>	1234567	4567890
<RelatesToMessageID>		1234567
<AdditionalMessageIndicator>		false
<MessageToFollowMessageIdentifier>		
<RxReferenceNumber>		
<PrescriberOrderNumber>		
	ClinicalInfoResponse from Prescriber	**Status from Pharmacy**
<MessageID>	ABC111	A77
<RelatesToMessageID>	1234567	4567890
<AdditionalMessageIndicator>	true	
<MessageToFollowMessageIdentifier>	4567890	
<RxReferenceNumber>		
<PrescriberOrderNumber>		

9.3.13 EXAMPLE 13

Pharmacy transmits an inquiry to the Central Fill Facility via a CFProductInquiry transaction.
The Central Fill Facility responds with a CFProductInquiryResponse. The <RelatesToMessageID> of the CFProductInquiryResponse transaction must contain the <MessageID> from the CFProductInquiry.

	CFProductInquiry from Pharmacy
Field Name	**Value**
<MessageID>	AG5687985
<RelatesToMessageID>	
<RxReferenceNumber>	
	CFProductInquiryResponse from Central Fill Facility
<MessageID>	9875954
<RelatesToMessageID>	AG5687985

9.3.14 EXAMPLE 14

Pharmacy requests an order fulfillment from the Central Fill Facility via a CFRxOrderRequest transaction.
The Central Fill Facility responds with a Status, and then sends subsequent information.
The Central Fill Facility sends a CFRxOrderCompletion. The <RelatesToMessageID> of the CFRxOrderCompletion transaction(s) must contain the <MessageID> from the CFRxOrderRequest.

(The RxReferenceNumber can be used as an additional tieback.)

	CFRxOrderRequest from Pharmacy	**CFRxOrderCompletion from Central Fill Facility**
Field Name	**Value**	
<MessageID>	AG5687564	9874582
<RelatesToMessageID>		AG5687564
<RxReferenceNumber>	8809612	8809612
	Status from Central Fill Facility	**Status from Pharmacy**
<MessageID>	98U76T5	AG99250
<RelatesToMessageID>	AG5687564	9874582
<RxReferenceNumber>	8809612	8809612

9.4 STATUS ELEMENTS

9.4.1 STATUS ELEMENTS NOTES

Status - <Code> value “010” (Successful – accepted by ultimate receiver) is to let the sender know that the message has been received by organization guaranteeing delivery. This value may only be used in a direct connect. It is not to be used in a mailbox environment.

9.5 REJECTION ELEMENTS

9.5.1 SYNTAX REJECTIONS

The Error transaction contains

<Code>
<DescriptionCode> (occurs up to 10 times)
<Description>

Most syntax errors are handled by the XML parser of the implementer. The syntax errors generated by the XML parser that are necessary to return to the sender of the transaction must contain

<Code> and **either** up to 10 <DescriptionCode>(s) **or** a <Description>, or both

A general XML syntax error must be avoided; identifying specific elements will help troubleshoot the problem.

The <Description> element is large enough to contain information directly from the parser. It is highly recommended that the xpath of the element in error be sent to assist the receiver in troubleshooting when appropriate.

The NCPDP ***External Code List*** contains <Code> that are to be used as appropriate to the syntax rejection condition.

9.5.2 APPLICATION REJECTIONS

Other transactions that support a denial situation (<Denied>/<DeniedNewPrescriptionToFollow>/etc.) such as RxChange, RefillResponse, CancelRxResponse, etc. contain

<ReasonCode> (occurs up to 10 times)
<DenialReason>
Up to 10 <ReasonCode>(s) **or** a <DenialReason>, or both

The NCPDP ***External Code List*** contains <ReasonCode> that are to be used as appropriate to the application rejection condition.

10. STANDARD CONVENTIONS

10.1 REPRESENTATION

There are two types of data representation: numeric, and alphanumeric. These are designated by a “n”, and “an”, respectively. When numbers follow this designation, they are the minimum length and the maximum length.

Representation	Explanation
an	alphanumeric that must have at least 1 nonblank character
n	numeric in the allowable values that must be sent with at least one valid value

Examples:

- an1..1 – alphanumeric field, minimum of 1 character, maximum of 1 character
- an1..3 – alphanumeric field, minimum of 1 character, maximum of 3 characters
- an1..15 – alphanumeric field, minimum of 1 character, maximum of 15 characters
- an – alphanumeric field, first character must be nonblank character, no constraint to length of the content (“string”)

- n1..1 – numeric field, minimum of 1 number, maximum of 1 number
- n1..9 – numeric field, minimum of 1 number, maximum of 9 numbers

In addition, the following representations are found in the schema:

Representation	Explanation
xsd:boolean	The type of an expression with two possible values, "True" and "False".
BooleanCode	NCPDP-defined backwards compatible type of expression with two possible values, “Y” and “N”.
DateTime	Format = CCYY-MM-DD THH:MM:SS
Date - Format	Format = CCYY-MM-DD

If the tag is marked as mandatory in the schema, the tag and a value in the tag must be sent. Empty tags must not be sent.

10.1.1 NUMERIC REPRESENTATION

A period is used to denote the decimal point. The decimal point must be counted when computing the maximum length of a data element. The decimal point should only be used when there are significant digits to the right of the decimal. It should not be used with whole numbers when a numeric designation. If the decimal point is necessary in a numeric designation, there must be at least one digit before and after the decimal point (i.e., 0.5).

Example. Consider the following possible formats for a 6 digit numeric field.

Recommended:	1.2345, 123.45, 12345, 0.1234, 1.2, 1234.5
Not Recommended:	.123, 12345., 1.00

10.1.2 DOLLAR REPRESENTATION

Dollar fields are alphanumeric. Dollar fields are always signed.

The decimal point should only be used when there are significant digits to the right of the decimal. It should not be used with whole numbers.

The sign if present is in the highest order significant digit of the dollar field. If negative, the – is used. If positive, no sign is used and does not occupy a position. The dollar sign is not sent.

Example: field with format of s$$$$$$$.cc (s = sign)

Positive:	21.6Ø, 123.45, .25, Ø.25
Negative:	-21.6Ø, -123.45, -Ø.25
Truncation:	Remove leading zeros

Not Recommended:	12345., -29., 21.6

10.1.3 Date or DateTime Representation

In XML, the Date fields contain the format of date and time, with hyphens to separate the subsets of the date. When time is used, a “T” separates the fields; and colons separate the subsets of the date and time.

Date is represented as YYYY-MM-DD
DateTime is represented as YYYY-MM-DDTHH:MM:SS
For example:

```
<SentTime>2010-10-01T08:15:22</SentTime>

<CensusEffectiveDate>2010-10-01</CensusEffectiveDate>

<DateOfBirth>
    <Date>1954-12-25</Date>
</DateOfBirth>

<SCRIPT:RequestedDates>
                <datatypes:StartDate>
                        <datatypes:Date>1967-08-13</datatypes:Date>
                        <datatypes:DateTime>2001-12-17T09:30:47Z</datatypes:DateTime>
                </datatypes:StartDate>
                <datatypes:EndDate>
                        <datatypes:Date>1967-08-13</datatypes:Date>
                        <datatypes:DateTime>2001-12-17T09:30:47Z</datatypes:DateTime>
                </datatypes:EndDate>
    </SCRIPT:RequestedDates>
```

10.1.4 Character Set

The following character set is used, unless trading partners determine a different set.

Letters, upper or lowercase	A to z
Numerals	0 to 9
Symbols	Printable characters

Alphabetic is defined as the character subset Letters, upper or lowercase (A-Z). Numeric is defined as the character subset Numerals (0-9). Alphanumeric is the character set Letters, upper and lowercase, Numerals, and Printable characters. Printable characters include, but are not limited to #!$%&*_- for example. Unprintable characters, such as control characters, are not used within the field sets. ½ ¼ are not allowable as these are symbols instead of characters.

Most symbols should be ‘safe’ within most alphanumeric fields; however, XML validation has issue with ‘less than’ < and with ampersand (&) and should be escaped. The most common escape encodings are:

ASCII Characters	XML Escape
quote (")	&quot;
apostrophe (')	&apos;

ampersand (&)	&amp;
less than (<)	&lt;
greater than (>)	&gt;

10.2 Truncation

All non-essential characters should be suppressed from the transaction where permissible (see section "*Representation*" above). Non-essential characters include leading spaces, leading zeroes, and trailing zeroes where permissible (see section "*Numeric Representation*" above).

The length of XML transaction can be optimized in several ways.

- Tags should not be transmitted unless they are required.
- Depending on the trading partner and the transaction, not all data elements will be utilized.
- Individual fields should not be padded with blanks or zeroes. Transmit only the characters used.

Although XML transactions can be truncated in several ways, software systems should not assume that a trading partner would always truncate. The systems should be capable of properly interpreting a full-length transaction.

10.3 Requirement Designation

10.3.1 Mandatory and Conditional Discussion

Each XML element has a requirement designation of either Mandatory or Conditional. Mandatory XML elements must be present in the transmission. Conditional XML elements may be omitted. If a conditional XML element is used, all mandatory XML elements contained within it are required.

The standard contains elements that are defined as conditional (having a minimum number of occurrences equal to 0). These elements are used for information that doesn't apply in every case, and so cannot be required to be populated. A simple example is <Patient><AddressLine2> is optional, but expected to be used when the patient has a secondary address line (for example lives in an apartment).

10.4 XML Element

Source: http://www.w3schools.com/xml/xml_elements.asp
An XML element is everything from (including) the element's start tag to (including) the element's end tag.
An element can contain other elements, simple text or a mixture of both. Elements can also have attributes.

```
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

In the example above, <bookstore> and <book> have **element contents**, because they contain other elements. <author> has **text content** because it contains text.
In the example above only <book> has an **attribute** (category="CHILDREN").

10.4.1 QUALIFIERS

If an element (a code/identifier) has a qualifier field associated with it, and that data element is to be used, the qualifier must be included. Qualifiers should only be sent if associated with an element.

10.5 TIME ZONE

In the Medication elements, <TimeZone> data elements are available for usage between the facility and the long term care pharmacy as part of the "Needed No Later Than" information. In addition, <TimeZone> is available for use when communicating drug administration events such as suspending medication administration, using the DrugAdministration transaction. Events communicated in the Census may include time as well as date, and the <TimeZone> element may be used to clarify those time values.

10.5.1 USAGE SCENARIOS

In the long term care environment, ordered medications are delivered to the facilities by the pharmacy, usually on a fixed delivery schedule. Because of this model, information is available for the facility and LTC pharmacy if the facility, when submitting a NewRx or Resupply, to indicate when the medication is needed. The pharmacy would then know if the order requires a special delivery or if it could go out with the next scheduled delivery. If both parties are in the same time zone, the <TimeZone> information is not needed. However, it may not always be apparent if the time zones are different between the sender and the receiver. It is therefore recommended that for consistency between the facility and the long term care pharmacy, when the <NeededNoLaterThanDate> information is sent, the <TimeZone> information is sent.

Likewise, when the prescriber/care setting notifies a pharmacy of an administration event using the DrugAdministration transaction, it is necessary to communicate the time of the suspension as well as the date. For example, the medication may need to be administered the morning a resident is scheduled for a medical procedure, but not in the evening. If the facility and pharmacy are in different time zones, it is recommended that <TimeZone> be used to clarify the suspend time.

It is recommended that <TimeZone> be used to clarify event times communicated in the Census transaction.

<TimeZoneDifferenceQuantity>
The <TimeZoneDifferenceQuantity> is HHMM offset from the Coordinated Universal Time (UTC) value "UT" (Universal Time Coordinate)).

Example	Description
-1000	10 hours from UTC = Hawaii-Aleutian Standard Time
-0900	9 hours from UTC = Alaska Standard Time
-0800	8 hours from UTC = Pacific Standard Time or Alaska Daylight Time
-0700	7 hours from UTC = Mountain Standard Time or Pacific Daylight Time
-0600	6 hours from UTC = Central Time or Mountain Daylight Time
-0500	5 hours from UTC = Eastern Standard Time or Central Daylight Time
-0400	4 hours from UTC = Eastern Daylight Saving Time
-0000	UTC without offset

The following are shown as examples.

10.5.2 EXAMPLE 1: FACILITY AND LONG TERM CARE PHARMACY IN SAME

TIME ZONE

The facility and long term care pharmacy are in the same time zone, Central Time. The facility requests delivery at a specific time with instructions. When the long term care pharmacy receives the request, the software interrogates the <TimeZone> information. Since the time zone is the same, no further calculation is needed.

Field Name	Remarks	Value	Comment
<NeededNoLaterThanDate>	Calendar date including time with minutes	200712150800	Requested delivery by 8 am on 12/15/2007 Delivery date/time, requested for (prior to and including) Delivery is requested to happen prior to or including the given date. CCYYMMDDHHMM C=Century; Y=Year; M=Month; D=Day; H=Hour; M=Minutes.
<TimeZoneIdentifier>	Defines the time zone used.	UT	Universal Time Coordinate
<TimeZoneDifferenceQuantity>	The time zone difference quantity is HHMM offset from the Coordinated Universal Time (UTC)	-0600	6 hours from UTC = Central Time or Mountain Daylight Time
<NeededNoLaterThanReason>	Free text additional instructions. For use with <NeededNoLaterThanDate> ; otherwise not used.	Please deliver early morning due to transfer of patient.	

10.5.3 EXAMPLE 2: FACILITY AND LONG TERM CARE PHARMACY IN DIFFERENT TIME ZONE

The facility and long term care pharmacy are in different time zones. The facility is in Central Time. In this example, the long term care pharmacy is in Eastern Time. The facility requests delivery at a specific time with instructions, based on their time zone. When the long term care pharmacy receives the request, the software interrogates the <TimeZone> information. Since the sender’s time zone is different than the receiver’s, further calculation is needed to correctly designate the delivery time to meet the sender’s request. Note that this example is exactly the same as Example 1; the difference is the perspective of the receiver – the time zone where the receiver is located.

Field Name	Remarks	Value	Comment
<NeededNoLaterThanDate>	Calendar date including time with minutes	200712150800	Requested delivery by 8 am on 12/15/2007 Delivery date/time, requested for (prior to and including) Delivery is requested to happen prior to or including the given date. CCYYMMDDHHMM C=Century; Y=Year; M=Month; D=Day; H=Hour; M=Minutes.
<TimeZoneIdentifier>	Defines the time zone used.	UT	Universal Time Coordinate
<TimeZoneDifferenceQuantity >	The time zone difference quantity is HHMM offset from the Coordinated Universal Time (UTC)	-0600	6 hours from UTC = Central Time or Mountain Daylight Time
<NeededNoLaterThanReason >	Free text additional instructions. For use with <NeededNoLaterThanDate> ; otherwise not used.	Please deliver early morning due to transfer of patient.	

11. TRANSMISSION EXAMPLES

11.1 EXAMPLE DATA ELEMENTS

In the examples, the following key data may be used:

Field Name	Value
Clinic ID	77777777
Clinic ID Qualifier	C
Clinic’s Password to Mailbox	PASSWORDA
Clinic’s Password to Pharmacy	PASSWORDQ
Clinic’s Password to Payer	PWPHY
Clinic’s Name	STATE CLINIC
Physician’s Name	MARK JONES
Physician’s State ID	6666666
Physician’s Address	211 CENTRAL ROAD, JONESVILLE, TN, 37777
Physician’s Telephone	6152219800
Physician’s Name	Tim Jonson
Physician’s State ID	3334444
Physician’s Address	151 EAST STREET, CLYDE, ME 11122
Physician’s Telephone	2078659191
Mailbox ID	111111
Mailbox ID Qualifier	M
Pharmacy ID	7701630
Pharmacy ID Qualifier	P
Pharmacy’s Password to Mailbox	PASSWORDB
Pharmacy’s Password to Clinic	PASSWORD5
Pharmacy’s Name	MAIN STREET PHARMACY
Pharmacy’s Address	5400 S 121 ST HALES CORNERS, TN 37122
Pharmacy Telephone	6152205656
Facility ID	FACILITY123
Facility’s Password to Payer	PASSWORDF
Payer ID	PAYER123
Payer’s Password to Pharmacy	PASSWORDZ
Payer’s Password to Facility	PASSWORDQ
Patient 1 ID	333445555
Patient 1 Name	MARY SMITH
Patient 1 Address	45 EAST ROAD SW, CLANCY, WI, 54999
Patient 1 Date of Birth	12/25/1954
Patient 2 ID	222222222
Patient 2 Name	JEFFREY TURNER
Patient 2 Address	1991 CELVIN ROAD, JONESVILLE, TN, 37777
Patient 2 Date of Birth	11/29/1957

Note that an ID and ID Qualifier are used to identify Mailbox as a recipient or sender of transactions. This identifier pair (111111:M for these examples) is used when transactions such as PasswordChange and GetMessage are sent. Additionally, responses to transactions where Mailbox intercedes will show Mailbox as the sender.

All requests and responses have two “tie back” fields. The first is <MessageID> which relates to the transmission. The second “tie back” field is <RelatesToMessageID> which is used to link transactions. See “*Trace Number Usage*” for usage rules.

11.2 EXAMPLE 1. INITIAL TRANSACTION TO A MAILBOX

The Mailbox or receiver may expect that the first transaction from a pharmacy or prescriber system will be used to establish the password for subsequent communication. This is accomplished by sending a PasswordChange transaction in which the old and new passwords are the same. The <OldPassword> and <NewPassword> are also the same. This password will then become the operative password. The example below indicates how this is performed.

Transaction
PasswordChange (from Pharmacy)
Status (from Mailbox)

PasswordChange (from Pharmacy)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 (http://www.altova.com)-->
<transport:Message StructuresVersion="String" ECLVersion="String" DatatypesVersion="String" TransactionDomain="SCRIPT" PA-StructuresVersion="String" TransactionVersion="String" TransportVersion="String" xsi:schemaLocation="http://www.ncpdp.org/schema/transport transport.xsd" xmlns:transport="http://www.ncpdp.org/schema/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <transport:Header>
                <transport:To Qualifier="M">111111</transport:To>
                <transport:From Qualifier="P">7701630</transport:From>
                <transport:MessageID>123561</transport:MessageID>
                <transport:SentTime>2010-10-01T08:15:22</transport:SentTime>
                <transport:Security>
                        <transport:UsernameToken>
                                <transport:Password Type="PasswordDigest">String</transport:Password>
                                <transport:Created>2001-12-17T09:30:47Z</transport:Created>
                        </transport:UsernameToken>
                        <transport:Sender>
                                <transport:SecondaryIdentification>PASSWORDB</transport:SecondaryIdentification>
                        </transport:Sender>
                </transport:Security>
                <transport:SenderSoftware>
                        <transport:SenderSoftwareDeveloper>ACE SOFTWARE</transport:SenderSoftwareDeveloper>
                        <transport:SenderSoftwareProduct>ACE1</transport:SenderSoftwareProduct>
                        <transport:SenderSoftwareVersionRelease>1.1</transport:SenderSoftwareVersionRelease>
                </transport:SenderSoftware>
        </transport:Header>
        <transport:Body>
                <transport:PasswordChange>
                        <transport:Request>
                                <transport:OldPassword>PASSWORDB</transport:OldPassword>
                                <transport:NewPassword>PASSWORDB</transport:NewPassword>
                        </transport:Request>
        </transport:PasswordChange>
        </transport:Body>
</transport:Message>
```

Notes:

Element	Value	Note
To	111111:M	Destination ID of Mailbox; M means Mailbox.
From	7701630:P	NCPDP Provider ID Number of pharmacy; P means pharmacy
MessageID	**123561**	**Pharmacy trace number for the transmission. Echoed back in the response message in RelatesToMessageID.**
RelatesToMessageID		**Not sent. Not applicable.**
SentTime	2010-10-01T08:15:22	10/01/2010 at 08:15:22
SecondaryIdentification	PASSWORDB	This is the normal location for a password in a transaction. At this point in time the pharmacy has no password established. This is the password that will ultimately be used.
SenderSoftwareDeveloper, SenderSoftwareProduct and SenderSoftwareVersionRelease	ACE SOFTWARE:ACE1:1.1	Sender Software Developer: ACE SOFTWARE Sender Software Product: ACE1 Sender Software Version Release: 1.1
TestMessage	1	Production
PasswordChange	PasswordChange	The transaction type: Password Change

Element	Value	Note
OldPassword, NewPassword	PASSWORDB PASSWORDB	The old and new passwords in that order. In the case of this message they are the same, since this establishing the password.

Mailbox Status Response to Pharmacy

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 (http://www.altova.com)-->
<transport:Message StructuresVersion="String" ECLVersion="String" DatatypesVersion="String" TransactionDomain="SCRIPT" PA-
StructuresVersion="String" TransactionVersion="String" TransportVersion="String" xsi:schemaLocation="http://www.ncpdp.org/schema/transport
transport.xsd" xmlns:transport="http://www.ncpdp.org/schema/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
		<transport:Header>
				<transport:To Qualifier="P">7701630</transport:To>
				<transport:From Qualifier="M">111111</transport:From>
				<transport:MessageID>66666666</transport:MessageID>
				<transport:RelatesToMessageID>123561</transport:RelatesToMessageID>
				<transport:SentTime>2010-10-01T08:15:23</transport:SentTime>
				<transport:SenderSoftware>
						<transport:SenderSoftwareDeveloper>SWITCH123</transport:SenderSoftwareDeveloper>
						<transport:SenderSoftwareProduct>ROUTEA</transport:SenderSoftwareProduct>
						<transport:SenderSoftwareVersionRelease>55</transport:SenderSoftwareVersionRelease>
				</transport:SenderSoftware>
		</transport:Header>
		<transport:Body>
				<transport:Status>
								<transport:Code>000</transport:Code>
		</transport:Status>
		</transport:Body>
</transport:Message>
```

Notes:

Per the rules, the Mailbox assigns their own trace number <MessageID> and echoes back the pharmacy trace number <RelatesToMessageID>.

Element	Value	Note
To	7701630:P	NCPDP Provider ID Number of pharmacy; P means pharmacy. Now the destination.
From	111111:M	Mailbox ID; M means Mailbox. Now the sender.
MessageID	**66666666**	**Mailbox trace number for this transmission.**
RelatesToMessageID	**123561**	**Pharmacy trace number is used to link the original message. Value in MessageID from request to the response.**
SentTime	2010-10-01T08:15:23	10/01/2010 at 08:15:23
SenderSoftwareDeveloper, SenderSoftwareProduct and SenderSoftwareVersion Release	SWITCH123:ROUTEA:55	Sender Software Developer: SWITCH123 Sender Software Product: ROUTEA Sender Software Version Release: 55 (For a STATUS response, these fields contain identification of the entity creating the response.)
Status	Status	The message type: Status
Status Code	000	The status type — 000 means no error.

11.3 EXAMPLE 2. PASSWORD CHANGE REQUEST

In this example, the Pharmacy is changing their password. Note that the destination is the Mailbox. The example would change the pharmacy's password from PASSWORDB to PASSWORDX. The pharmacy system should always check for mail at the Mailbox when a connection is made for any reason. In this example, no mail is present.

Transaction
PasswordChange (from Pharmacy)
Status (from Mailbox)
GetMessage (from Pharmacy)
Status (No More Mail, from Mailbox)

PasswordChange (from Pharmacy)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 (http://www.altova.com)-->
```

```
<transport:Message StructuresVersion="String" ECLVersion="String" DatatypesVersion="String" TransactionDomain="SCRIPT" PA-StructuresVersion="String" TransactionVersion="String" TransportVersion="String" xsi:schemaLocation="http://www.ncpdp.org/schema/transport transport.xsd" xmlns:transport="http://www.ncpdp.org/schema/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <transport:Header>
        <transport:To Qualifier="M">111111</transport:To>
        <transport:From Qualifier="P">7701630</transport:From>
        <transport:MessageID>123562</transport:MessageID>
        <transport:SentTime>2010-10-01T08:15:22</transport:SentTime>
        <transport:Security>
            <transport:UsernameToken>
                <transport:Password Type="PasswordDigest">String</transport:Password>
                <transport:Created>2001-12-17T09:30:47Z</transport:Created>
            </transport:UsernameToken>
            <transport:Sender>
                <transport:SecondaryIdentification>PASSWORDB</transport:SecondaryIdentification>
            </transport:Sender>
        </transport:Security>
        <transport:SenderSoftware>
            <transport:SenderSoftwareDeveloper>ACE SOFTWARE</transport:SenderSoftwareDeveloper>
            <transport:SenderSoftwareProduct>ACE1</transport:SenderSoftwareProduct>
            <transport:SenderSoftwareVersionRelease>1.1</transport:SenderSoftwareVersionRelease>
        </transport:SenderSoftware>
    </transport:Header>
    <transport:Body>
        <transport:PasswordChange>
            <transport:Request>
                <transport:OldPassword>PASSWORDB</transport:OldPassword>
                <transport:NewPassword>PASSWORDX</transport:NewPassword>
            </transport:Request>
    </transport:PasswordChange>
</transport:Body>
</transport:Message>
```

Notes:

Element	Value	Note
To	111111:M	Destination ID of Mailbox
From	7701630:P	NCPDP Provider ID Number of pharmacy; P means pharmacy
MessageID	**123562**	**Pharmacy trace number for the transmission. Echoed back in the response message RelatesToMessageID.**
RelatesToMessageID		**Not sent. Not applicable.**
SentTime	2010-11-01T08:15:22	Date and time transaction was sent 11/01/2010 08:15:22 AM
SecondaryIdentification	PASSWORDB	The current password.
SenderSoftwareDeveloper, SenderSoftwareProduct and SenderSoftwareVersionRelease	ACE SOFTWARE:ACE1:1.1	Sender Software Developer: ACE SOFTWARE Sender Software Product: ACE1 Sender Software Version Release: 1.1
TestMessage	1	Production
PasswordChange	PasswordChange	The transaction type: Password Change
OldPassword	PASSWORDB	The old password.
NewPassword	PASSWORDX	The new password

Mailbox Status Response to Pharmacy

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 (http://www.altova.com)-->
<transport:Message StructuresVersion="String" ECLVersion="String" DatatypesVersion="String" TransactionDomain="SCRIPT" PA-StructuresVersion="String" TransactionVersion="String" TransportVersion="String" xsi:schemaLocation="http://www.ncpdp.org/schema/transport transport.xsd" xmlns:transport="http://www.ncpdp.org/schema/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <transport:Header>
        <transport:To Qualifier="P">7701630</transport:To>
        <transport:From Qualifier="M">111111</transport:From>
        <transport:MessageID>66666677</transport:MessageID>
        <transport:RelatesToMessageID>123562</transport:RelatesToMessageID>
        <transport:SentTime>2010-10-01T08:15:23</transport:SentTime>
        <transport:SenderSoftware>
            <transport:SenderSoftwareDeveloper>SWITCH123</transport:SenderSoftwareDeveloper>
            <transport:SenderSoftwareProduct>ROUTEA</transport:SenderSoftwareProduct>
            <transport:SenderSoftwareVersionRelease>55</transport:SenderSoftwareVersionRelease>
        </transport:SenderSoftware>
    </transport:Header>
    <transport:Body>
        <transport:Status>
```

```
                                        <transport:Code>000</transport:Code>
                    </transport:Status>
          </transport:Body>
</transport:Message>
```

Notes:

Per the rules the Mailbox assigns their own trace number <MessageID> and echoes back the pharmacy trace number <RelatesToMessageID>.

Element	Value	Note
To	7701630:P	NCPDP Provider ID Number of pharmacy; P means pharmacy. Now the destination.
From	111111:M	Mailbox ID. Now the sender.
MessageID	**66666677**	**Mailbox trace number for this transmission.**
RelatesToMessageID	**123562**	**Pharmacy trace number is used to link the original transaction (value in MessageID) from request to the response.**
SenderSoftwareDeveloper, SenderSoftwareProduct and SenderSoftwareVersionRelease	SWITCH123:ROUTEA:55	Sender Software Developer: SWITCH123 Sender Software Product: ROUTEA Sender Software Version Release: 55 (For a STATUS response, these fields contain identification of the entity creating the response.)
Status	Status	The transaction type: Status
Status Code	000	The status type — 000 means no error.

GetMessage (from Pharmacy)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 (http://www.altova.com)-->
<transport:Message StructuresVersion="String" ECLVersion="String" DatatypesVersion="String" TransactionDomain="SCRIPT" PA-StructuresVersion="String" TransactionVersion="String" TransportVersion="String" xsi:schemaLocation="http://www.ncpdp.org/schema/transport transport.xsd" xmlns:transport="http://www.ncpdp.org/schema/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
          <transport:Header>
                    <transport:To Qualifier="M">111111</transport:To>
                    <transport:From Qualifier="P">7701630</transport:From>
                    <transport:MessageID>123563</transport:MessageID>
                    <transport:SentTime>2010-10-01T08:15:24</transport:SentTime>
                    <transport:Security>
                              <transport:UsernameToken>
                                        <transport:Password Type="PasswordDigest">String</transport:Password>
                                        <transport:Created>2001-12-17T09:30:47Z</transport:Created>
                              </transport:UsernameToken>
                              <transport:Sender>
                                        <transport:SecondaryIdentification>PASSWORDX</transport:SecondaryIdentification>
                              </transport:Sender>
                    </transport:Security>
                    <transport:SenderSoftware>
                              <transport:SenderSoftwareDeveloper>ACE SOFTWARE</transport:SenderSoftwareDeveloper>
                              <transport:SenderSoftwareProduct>ACE1</transport:SenderSoftwareProduct>
                              <transport:SenderSoftwareVersionRelease>1.1</transport:SenderSoftwareVersionRelease>
                    </transport:SenderSoftware>
          </transport:Header>
          <transport:Body>
                    <transport:GetMessage>
                    </transport:GetMessage>
          </transport:Body>
</transport:Message>
```

Notes:

Element	Value	Note
MessageID	**123563**	**Pharmacy trace number for the transmission. Echoed back in the response transaction in RelatesToMessageID.**
RelatesToMessageID		**Not sent. Not applicable.**
SecondaryIdentification	PASSWORDX	Now the current password.
GetMessage	GetMessage	The transaction type: Get Message
SentTime	2010-10-01T08:15:24	11/01/2010 at 08:15:24 am.
SenderSoftwareDeveloper, SenderSoftwareProduct and SenderSoftwareVersionRelease	ACE SOFTWARE:ACE1:1.1	Sender Software Developer: ACE SOFTWARE Sender Software Product: ACE1 Sender Software Version Release: 1.1

Status (No More Mail, from Mailbox)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 (http://www.altova.com)-->
<transport:Message StructuresVersion="String" ECLVersion="String" DatatypesVersion="String" TransactionDomain="SCRIPT" PA-
StructuresVersion="String" TransactionVersion="String" TransportVersion="String" xsi:schemaLocation="http://www.ncpdp.org/schema/transport
transport.xsd" xmlns:transport="http://www.ncpdp.org/schema/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <transport:Header>
                <transport:To Qualifier="P">7701630</transport:To>
                <transport:From Qualifier="M">111111</transport:From>
                <transport:MessageID>66666679</transport:MessageID>
                <transport:RelatesToMessageID>123563</transport:RelatesToMessageID>
                <transport:SentTime>2010-10-01T08:15:23</transport:SentTime>
                <transport:SenderSoftware>
                        <transport:SenderSoftwareDeveloper>SWITCH123</transport:SenderSoftwareDeveloper>
                        <transport:SenderSoftwareProduct>ROUTEA</transport:SenderSoftwareProduct>
                        <transport:SenderSoftwareVersionRelease>55</transport:SenderSoftwareVersionRelease>
                </transport:SenderSoftware>
        </transport:Header>
        <transport:Body>
                <transport:Status>
                                <transport:Code>002</transport:Code>
        </transport:Status>
        </transport:Body>
</transport:Message>
```

Notes:

Per the rules the Mailbox assigns their own trace number <MessageID> and echoes back the pharmacy trace number <RelatesToMessageID>.

Element	Value	Note
MessageID	**66666679**	**Mailbox trace number for this transmission.**
RelatesToMessageID	**123563**	**Pharmacy trace number is used to link the original transaction (value in MessageID) from request to the response.**
SentTime	2010-10-01T08:15:24	11/01/2010 at 08:15:24 am.
SenderSoftwareDeveloper, SenderSoftwareProduct and SenderSoftwareVersionRelease	SWITCH123:ROUTEA:55	Sender Software Developer: SWITCH123 Sender Software Product: ROUTEA Sender Software Version Release: 55 (For a STATUS response, these fields contain identification of the entity creating the response.)
Status	Status	Status response to a GetMessage request means there is no more mail.
Status Code	002	The status type — 002 means no error and no more mail.

11.4 EXAMPLE 3. ERROR - SENDER IS UNKNOWN TO MAILBOX

In this example, the pharmacy initiates the **first** transaction to a Mailbox - a request for a refill authorization. The Mailbox responds with an Error that the pharmacy is unauthorized for access (the PasswordChange transaction should have been sent first). **This is just an example of a simpler error situation.**

Transaction
RefillRequest (from Pharmacy)
Error (from Mailbox)

For the purposes of this example, assume a paper/fax prescription, so no <RelatesToMessageID>.

The pharmacy has assigned prescription number PH456 on their system.

RefillRequest (from Pharmacy)

For this example, only the pertinent identifiers and passwords are shown.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 (http://www.altova.com)-->
<!--This transaction will not validate as the RefillRequest information was intentionally left out to just show the error situation.-->
<transport:Message StructuresVersion="String" ECLVersion="String" DatatypesVersion="String" TransactionDomain="SCRIPT" PA-StructuresVersion="String" TransactionVersion="String" TransportVersion="String" xsi:schemaLocation="http://www.ncpdp.org/schema/transport transport.xsd" xmlns:transport="http://www.ncpdp.org/schema/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <transport:Header>
                <transport:To Qualifier="C">77777777</transport:To>
                <transport:From Qualifier="P">7701630</transport:From>
                <transport:MessageID>A55</transport:MessageID>
                <transport:SentTime>2010-10-01T08:15:22</transport:SentTime>
                <transport:Security>
                        <transport:UsernameToken>
                                <transport:Password Type="PasswordDigest">String</transport:Password>
                                <transport:Created>2001-12-17T09:30:47Z</transport:Created>
                        </transport:UsernameToken>
                        <transport:Sender>
                                <transport:SecondaryIdentification>PASSWORDB</transport:SecondaryIdentification>
                        </transport:Sender>
                </transport:Security>
                <transport:SenderSoftware>
                        <transport:SenderSoftwareDeveloper>ACE SOFTWARE</transport:SenderSoftwareDeveloper>
                        <transport:SenderSoftwareProduct>ACE1</transport:SenderSoftwareProduct>
                        <transport:SenderSoftwareVersionRelease>1.1</transport:SenderSoftwareVersionRelease>
                </transport:SenderSoftware>
                <transport:RxReferenceNumber>PH456</transport:RxReferenceNumber>
        </transport:Header>
        <transport:Body>
                <transport:RefillRequest>
<!--detail left out intentionally for this example-->
                </transport:RefillRequest>
        </transport:Body>
</transport:Message>
```

Notes:

Element	Value	Note
To	77777777:C	This is the Clinic ID of the receiver; C means it is a Clinic.
From	7701630:P	NCPDP Provider ID Number of pharmacy; P means it is a pharmacy. This is the sender. It must be the pharmacy ID.
MessageID	**A55**	**Pharmacy system trace number for the transmission. Echoed back in the response transaction in RelatesToMessageID.**
RelatesToMessageID		**Not sent as original prescription was paper/fax.**
SentTime	2010-11-01T10:15:22	Date and time transaction was sent 11/01/2010 10:15:22 A.M.
SecondaryIdentification	PASSWORDB	This is the password of the pharmacy on the Mailbox's system but is incorrect.
SenderSoftwareDeveloper, SenderSoftwareProduct and SenderSoftwareVersionRelease	ACE SOFTWARE:ACE1:1.1	Sender Software Developer: ACE SOFTWARE Sender Software Product: ACE1 Sender Software Version Release: 1.1
RefillRequest	RefillRequest	The transaction type: Refill Request.
RxReferenceNumber	**PH456**	**This is the prescription number assigned by the pharmacy system.**
PrescriberOrderNumber		**Not sent as original prescription was paper/fax.**

Error (from Mailbox)

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 (http://www.altova.com)-->
<transport:Message StructuresVersion="String" ECLVersion="String" DatatypesVersion="String" TransactionDomain="SCRIPT" PA-StructuresVersion="String" TransactionVersion="String" TransportVersion="String" xsi:schemaLocation="http://www.ncpdp.org/schema/transport transport.xsd" xmlns:transport="http://www.ncpdp.org/schema/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <transport:Header>
                <transport:To Qualifier="P">7701630</transport:To>
                <transport:From Qualifier="M">111111</transport:From>
                <transport:MessageID>2222</transport:MessageID>
                <transport:RelatesToMessageID>A55</transport:RelatesToMessageID>
                <transport:SentTime>2010-10-01T08:15:23</transport:SentTime>
                <transport:SenderSoftware>
                        <transport:SenderSoftwareDeveloper>SWITCH123</transport:SenderSoftwareDeveloper>
                        <transport:SenderSoftwareProduct>ROUTEA</transport:SenderSoftwareProduct>
                        <transport:SenderSoftwareVersionRelease>55</transport:SenderSoftwareVersionRelease>
                </transport:SenderSoftware>
        </transport:Header>
        <transport:Body>
                <transport:Error>
                                <transport:Code>900</transport:Code>
                                <transport:DescriptionCode>1000</transport:DescriptionCode>
                                <transport:Description>parser message of invalid sender</transport:Description>
        </transport:Error>
        </transport:Body>
</transport:Message>
```

Notes:

Element	Value	Note
MessageID	**2222**	**Mailbox trace number for the transmission.**
RelatesToMessageID	**A55**	**Pharmacy system trace number is used to link the request to the response.**
Error	Error	Error response to a request
Code	900	Transaction was rejected
DescriptionCode	1000	Unable to identify based on the information submitted
Description	*parser message and/or application error*	Mailbox text explanation of error

11.5 Example 4. Verify

For examples of Verify transactions, see the NCPDP ***SCRIPT Implementation Guide***. Below is an example of the Verify out of context, to see the structure.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2010 (http://www.altova.com)-->
<transport:Message StructuresVersion="String" ECLVersion="String" DatatypesVersion="String" TransactionDomain="SCRIPT" PA-StructuresVersion="String" TransactionVersion="String" TransportVersion="String" xsi:schemaLocation="http://www.ncpdp.org/schema/transport transport.xsd" xmlns:transport="http://www.ncpdp.org/schema/transport" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <transport:Header>
                <transport:To Qualifier="P">7701630</transport:To>
                <transport:From Qualifier="C">77777777</transport:From>
                <transport:MessageID>A30</transport:MessageID>
                <transport:RelatesToMessageID>1234567</transport:RelatesToMessageID>
                <transport:SentTime>2010-10-01T08:20:30</transport:SentTime>
                <transport:SenderSoftware>
                        <transport:SenderSoftwareDeveloper>ACE SOFTWARE</transport:SenderSoftwareDeveloper>
                        <transport:SenderSoftwareProduct>ACE1</transport:SenderSoftwareProduct>
                        <transport:SenderSoftwareVersionRelease>1.1</transport:SenderSoftwareVersionRelease>
                </transport:SenderSoftware>
                <transport:PrescriberOrderNumber>110088</transport:PrescriberOrderNumber>
        </transport:Header>
        <transport:Body>
                <transport:Verify></transport:Verify>
        </transport:Body>
</transport:Message>
```

12. FREQUENTLY ASKED QUESTIONS

12.1 [QUESTION]

[Response]

13. UPDATES AND CORRECTIONS TO STANDARDS

The Data Element Request Form (DERF) provides the mechanism for changing NCPDP standards and using or requesting new data elements and new code set values in business functions. To request a change in NCPDP standards, complete an NCPDP Data Element Request Form, available at www.ncpdp.org. Appropriate NCPDP Work Groups and Committees consider information submitted on the DERF. The Data Element Request Form process makes it possible for NCPDP working committees to adequately address these concerns before accepting or approving new information requests into a standard. The final acceptance of new requests into the standard is made by NCPDP at the suggestion or recommendation of the Work Group or Committee, and must be approved by consensus or consensus ballot of the membership.

14. APPENDIX A. HISTORY OF STANDARD CHANGES

14.1 VERSION 2Ø11Ø71

The following Query transactions have been added.

- NewRxRequest
- NewRxResponseDenied
- ClinicalInfoRequest
- ClinicalInfoResponse

Clarification and examples of the use of trace numbers for these transactions was added to section "*Trace Number Usage*".

14.2 VERSION 2Ø12Ø31

Previous versions of data types allowed for empty XML tags (an..35) or non-empty XML tags an..35M. Empty tags cause issues with mandatory fields where an empty tag satisfies the requirement of sending the field. Eliminating the empty tags will allow the schema to enforce mandatory requirements and stop confusion around the data types. It will also result in fewer data types. Section "*Representation*" has been modified. The original appears below.

> There are three types of data representation: alphabetic, numeric, and alphanumeric. These are designated by a "n", and "an", respectively. If a number follows this designation (i.e., an3), this means the data value must be that length (i.e., the data value must be 3 alphanumeric bytes). If the designation is followed by ".." (i.e., an..3), this means the data value can be up to that length (i.e., up to 3 alphanumeric bytes).

Representation	Explanation
an	alphanumeric character set in the allowable values is supported
anM	alphanumeric that must have at least 1 nonblank character
n	numeric set in the allowable values is supported
nM	numeric in the allowable values that must be sent with at least one valid value

> If the tag is marked as mandatory in the schema, **the tag and a value in the tag must be sent**.

The "Eliminating empty tags" rationale in previous paragraph also applies to all elements that use the primitive datatype "string". These elements have been changed to utilize "an" (section "*Representation*") as their datatypes.
For example:

```
<xsd:element name="AttachmentSource" type="xsd:string">
```

has been changed to

```
<xsd:element name="AttachmentSource" type="datatypes:an">
```

Section "*Numeric Representation*" was modified to be clearer in the example.

From
Example. Consider the following possible values for a 5 digit field.
To
Example. Consider the following possible formats for a 5 digit numeric field.

In previous versions ID types for entities (e.g., Prescriber, Pharmacy, Patient, Payer, Facility and Medication (Prior Authorization)) not all the values applied to all types (e.g., a patient would not be identified with a payer id or bin location number.) Each entity now has a separate list so the logical values can enforced through the schema. Also changed is the ID type from a choice list to a sequence so mandatory tags like the NPI and NCPDP ID can be enforced per the Implementation Guide. Medication Prior Authorization ID type now matches the other id types since it can be used for the Promotion Number and Prescriber Order Number.

The <PlaceLocationQualifier> has been sunsetted.

14.1 VERSION 2Ø13Ø11

To allow for additional identifiers, including but not limited to, standard email addresses in the <From> and <To> field of SCRIPT and Specialized transactions, the validation has been relaxed to (mailto:)?[^\.]{1,35}(\.[^@]{1,3})?(@[A-Za-z0-9]+\.[A-Za-z]{2,4})? to allow for any alphanumeric string up to 80 characters.

This version, in the accompanying implementation guides, supports standardized demographic elements.

14.2 VERSION 2Ø13Ø41

Since <ZIPCode> element name change to <PostalCode>, the reference in section "*Document Scope*" was modified.

14.3 VERSION 2Ø13Ø71

Section "*Updates and Corrections to Standards*" was added.

An additional paragraph was added to "*Mandatory and Conditional Discussion*" to provide an example (<AddressLine2>).

An error was corrected in section "*Representation*". In this table

In addition, the following representations are found in the schema:

Representation	Explanation
xsd:boolean	the type of an expression with two possible values, "true" and "false".
BooleanCode	NCPDP-defined backwards compatible type of expression with two possible values, "T" and "F". – Should be "Y" and "N".
DateTime	Format = CCYY-MM-DD THH:MM:SS
Date - Format	Format = CCYY-MM-DD

The Prior Authorization transactions were added. The pa-structures.xsd was added to the schemas. "SCRIPT" to "script" was changed in the namespace prefix of the script.xsd to be consistent with other xsd e.g., transport, datatypes. The URI was changed from "SCRIPT" to "script" and "Transport" to "transport" to be consistent. The examples were updated.

- PAInitiationRequest
- PAInitiationResponse
- PARequest
- PAResponse
- PAAppealRequest
- PAAppealResponse
- PACancelRequest
- PACancelResponse

Added support for Central Fill Transactions which include:

- CFInventoryList
- CFProductInquiry
- CFProductInquiryResponse
- CFRxOrderRequest
- CFRxOrderCompletion
- CFRxOrderCancel
- CFManifest

Information was added for the use of Dollar fields in section "*Dollar Representation*".

It was clarified that the Status transaction can be used as a response to an Error.