



Implementation Guide for Expressing Context in Direct Messaging

Version 1.1

19 June 2018

Contents

Status of this Guide	4
Introduction	4
Overview	4
Assumptions.....	4
Requirements	4
1.0 The X-Direct-Context Header.....	5
2.0 Metadata: MIME Headers and Encoding	5
3.0 Metadata: Content	6
3.1 Version	6
3.2 Transaction ID	6
3.3 Patient ID.....	7
3.4 Transaction Type	8
3.5 Purpose of Use	8
3.6 Patient Matching Parameters	9
3.7 Encapsulation of Other Data Types	10
4.0 Recipient Responses: Error Conditions	11
4.1 Version Not Supported	11
4.2 Transaction Type Not Supported.....	11
4.3 Errors Consuming Content	11
5.0 Security Considerations	12
6.0 Acknowledgements	12
7.0 References.....	13
Appendix A Examples	14
A.1 Context for a PDF document attachment	14
A.2 Context for an encapsulated HL7 message	15
A.3 Error notification message.....	15

Change Control

Date	Version	Description of changes
06-20-2016	0.1	Initial Draft
10-18-2016	1.0-DRAFT-2016101801	Addresses feedback from Direct Project Implementers Workgroup.
12-29-2016	1.0-DRAFT-2016122901, Draft for Trial Use	Incorporates final set of feedback from Direct Project Implementers Workgroup (IWG) gathered in October 2016 meeting. Per IWG, this draft is offered for trial use and additional feedback prior to further advancement.
06-19-2018	1.1	Incorporates additional feedback from IWG gathered during trial use period. Skips "1.0" version designation and advances to "1.1" since "1.0 Draft for Trial Use" has been in use in the field long enough to be considered the de facto "1.0" publication.

Status of this Guide

This document is PUBLISHED.

Introduction

Overview

This document defines an extensible mechanism to express the context of a Direct message exchange by providing a framework for the inclusion of contextual metadata by the message sender. Such metadata may be helpful for routing and processing of Direct messages, such as in cases where the payload format contains limited or no intrinsic metadata, or when the reason for the transmission may not be easily determined from the payload content.

The framework was designed so that the metadata could also be available to users of non-conforming applications in a human-readable format. A separate attachment containing the payload metadata was selected over a series of [RFC 822](#) headers, as such headers may not be as easily viewed by all Mail User Agents (MUAs). Similarly, human-readable metadata parameter names and values were selected over Object Identifiers (OIDs) or other non-human-readable representations.

Assumptions

It is assumed that any Direct message (including any mail system reports such as Delivery Status Notifications), header, or attachment described in this document conforms to the requirements of Direct Project's [Applicability Statement for Secure Health Transport v1.2](#), referred to as "[Applicability Statement](#)" in subsequent sections.

Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#).

An implementation is not compliant if it fails to satisfy one or more of the MUST, SHALL, or REQUIRED level requirements for the protocols it implements. An implementation that satisfies all the MUST, SHALL, or REQUIRED level and all the SHOULD level requirements for its protocols is said to be "unconditionally compliant"; one that satisfies all the MUST, SHALL, or REQUIRED level requirements but not all the SHOULD level requirements for its protocols is said to be "conditionally compliant."

In addition, annotations called “*Implementation Note:*” are used to provide additional clarification to implementers. These are non-normative and provided for clarification and informational purposes only.

1.0 The X-Direct-Context Header

A sending system SHALL indicate its conformance to this guide by including the following header with the [RFC 822](#) headers of the Health Content Container (as defined by the [Applicability Statement](#)):

```
X-Direct-Context: <cid-identifier>
```

The MIME body of the conforming Health Content Container SHALL be a MIME multipart container. The `cid-identifier` in the `X-Direct-Context` header SHALL correspond to the value of a `Content-ID` header of the MIME part that contains the Metadata attachment (see [Section 2.0](#) of this document), as per [RFC 2045](#). The MIME part referenced by the `Content-ID` header SHALL NOT be nested within another MIME multipart container. Only a single context is permitted per message. When multiple attachments are included by a sender, the same context SHALL apply to all attachments.

2.0 Metadata: MIME Headers and Encoding

A sending system SHALL include an attachment within a conforming Direct message that contains the payload metadata. This MIME part SHALL be identified by a `Content-ID` header field with a value equal to the `cid-identifier` specified in the `X-Direct-Context` header as defined in [Section 1.0](#) of this document.

In order to permit implementations that do not conform to this guide to provide the metadata in a human readable form to operators, the MIME part MUST contain the following headers:

```
Content-Type: text/plain
Content-ID: <cid-identifier>
Content-Disposition: attachment; filename=metadata.txt
```

Line breaks in this attachment must be canonicalized to `<CR><LF>` in accordance with section 4.1.1 of [RFC 2046](#). If the attachment contains any 8-bit-wide words (such as with the use of certain UTF-8 characters), content encoding to 7-bit-wide words MUST be applied as per section 6 of [RFC 2045](#).

3.0 Metadata: Content

The content of the metadata attachment SHALL be structured as a series of one or more metadata elements expressed as parameter/value pairs, separated by line breaks, as defined further in the following subsections. Lines that begin with whitespace shall be treated as continuations of the preceding line. The ABNF notation used in this guide conforms to [RFC 5234](#).

```
metadata = 1*(metadata-element CRLF)
metadata-element = metadata-parameter ":" metadata-value
metadata-parameter = <printable ASCII characters other than
whitespace and ":">
metadata-value = <printable ASCII characters with additional
restrictions as appropriate for each header; leading or
trailing whitespace is ignored>
```

Each metadata parameter MAY specify additional requirements for allowed metadata values, and MAY specify cardinality requirements on the parameter itself. Metadata parameter names are case-insensitive. Metadata values are case-insensitive, unless specified otherwise in this guide. All metadata-elements listed below are OPTIONAL, unless otherwise specified.

A conforming system MUST understand the `version-element`, and MUST conform to any requirements below where metadata must be echoed in response messages. Otherwise, a receiving system MAY use the enclosed metadata in any manner it chooses, including ignoring the metadata or subsets of the metadata.

3.1 Version

```
version-element = "version:" version-identifier
version-identifier = "1.1"
```

The metadata MUST include one `version-element`. Future versions of this document that introduce new functionality that is not backwards compatible with this version SHALL increase the numeric value of the `version-identifier`.

3.2 Transaction ID

```
id-element = "id:" unique-identifier
```

The `id-element` defines a unique and persistent identifier assigned by the originating party to uniquely identify a sequence of related transactions, such as a query and response or an order for a diagnostic test and its result. This `id-element` also correlates any error reporting or required acknowledgments when this is not already encapsulated in the message content. The `unique-identifier` is case-sensitive. The `unique-identifier` MAY be the same as the `Message-ID` of the message if appropriate. Only one `id-element` is permitted in the

metadata. When responding to a message that contains this element, the same `id`-element SHALL be included in the response metadata.

Examples:

- `id: <2142848@direct.example.com>`
- `id: 2ba8a9a1-0f59-4688-b818-67930ae26979`

Implementation Note: If angle brackets are included in the `unique-identifier` (as in the first example in this section), then these angle brackets are considered an integral part of the identifier and should not be removed in response metadata, i.e. `<abc>` is not equivalent to `abc`.

3.3 Patient ID

```
patient-id-element = "patient-id:" pid-instance *("; " pid-
instance)
pid-instance = pid-context ":" local-patient-id
pid-context = <Assigning Authority Domain ID or UUID as
described in text below>
local-patient-id = <printable ASCII characters other than
whitespace and ";">
```

The sender of a Direct message who wishes to identify the patient identifier in its local context SHALL construct a `pid-instance` using its unique Assigning Authority OID and its local patient identifier. The `pid-context` value is a globally unique value for each issuer of patient identifiers. If the sender has an Assigning Authority Domain ID (AA) that it uses for transactions under the Integrating the Healthcare Enterprise Information Technology Infrastructure Technical Framework (IHE ITI TF), then the sender SHOULD use that ID as the `pid-context`. If such an AA is not used, then the sender MUST use a Name-Based Universally Unique Identifier (UUID) as defined in Section 4.3 of [RFC 4122](#) using the sender's Direct Address or Direct Domain as the input, with the resulting 16 octet UUID value expressed as an unsigned integer as the final value of an OID in the 2.25 OID arc (e.g., a UUID with an unsigned integer value of 123456789 would be expressed as 2.25.123456789). At most one `patient-id-element` is permitted in the metadata.

All `pid-instances` are intended to represent the same patient in the corresponding `pid-context`, thus each context may correspond to at most one patient. Only one `pid-instance` is permitted per `pid-context`, i.e. a `pid-context` MUST NOT appear more than once in a single `patient-id-element`. A party MAY add, modify, or remove its own `pid-instance` when constructing the `patient-id-element` in responses. Otherwise, each `pid-instance` included in an incoming message MUST be included in the response. Each participant SHOULD include its preferred patient identifier as its `local-patient-id`.

Example:

```
patient-id: 2.16.840.1.113883.19.999999:123456;  
2.16.840.1.113883.19.888888:75774
```

3.4 Transaction Type

```
type-element = "type:" category "/" action  
category = "laboratory"  
    / "radiology"  
    / "pharmacy"  
    / "referral"  
    / "general"  
    / "error"  
action = "order"  
    / "report"  
    / "result"  
    / "request"  
    / "response"  
    / "notification"
```

When a `type-element` is included in the metadata, the `category` is intended to identify the workflow, and the `action` is intended to identify the role of the message sender in the transaction sequence, i.e., a request action would generally be followed by a response action.

Example:

```
type: radiology/report
```

Implementation Note: The above vocabulary for `category` and `action` values is normative, i.e., other values are not permitted.

3.5 Purpose of Use

```
purpose-element = "purpose:" purpose-value  
purpose-value = "treatment" / "payment" / "operations" /  
"emergency" / "research"
```

When a message sender requests the disclosure of healthcare information from the recipient, the `purpose-element` identifies the purpose for which the sender will use the disclosed information.

3.6 Patient Matching Parameters

```
patient-data-element = "patient:" patient-attribute *(";"  
patient-attribute)  
patient-attribute = patient-parameter-name "=" patient-  
parameter-value  
patient-parameter-name = "givenName"  
    / "surname"  
    / "middleName"  
    / "dateOfBirth"  
    / "gender"  
    / "socialSecurityNumber"  
    / "telephoneNumber"  
    / "streetAddress"  
    / "localityName"  
    / "stateOrProvinceName"  
    / "postalCode"  
    / "country"  
    / "directAddress"
```

patient-parameter-value = <based on patient-parameter, see text below for additional details and restrictions>

Both the patient-parameter-name and patient-parameter-value are case insensitive. The patient-parameter-value format is defined based on the patient-parameter-name and MUST NOT contain the ";", CR, or LF characters. Contiguous whitespace MUST be treated as a single space. Leading and trailing whitespace MUST be ignored. A patient-parameter-name MUST NOT appear more than once in the patient-data-element. When a parameter is included, the following additional requirements apply:

- dateOfBirth: MUST be in YYYY-MM-DD or YYYY format (if month/day are unknown).
- socialSecurityNumber: MUST contain 9 digits or last 4 digits; MAY include hyphens.
- telephoneNumber: For telephone numbers in the United States of America and Canada, MUST contain 10 digits, including area code; MAY include parentheses or hyphens. For telephone numbers in other countries, MUST contain a telephone number formatted as per the ITU-T [E.164](#) standard, which can be identified by a leading "+" character.
- localityName: MUST contain the full locality name; the locality name MUST NOT be abbreviated.
- stateOrProvinceName: MUST contain the full state or province name or the state or province abbreviation portion of the corresponding [ISO 3166-2](#) code. For example, Alabama, USA = "AL" and Sonora, Mexico = "SON".
- postalCode: For locations within the United States of America, MUST contain 5 digit or 9 digit zip code; for 9 digit zip codes, MAY include a hyphen between the first 5 and last 4 digits.

- `middleName`: MUST contain middle name or first letter of middle name; MAY include first letter plus a period.
- `country`: MUST contain the two-letter [ISO 3166](#) country code. If the country code is omitted, then the other patient matching parameters relating to a location, if any, SHALL be interpreted as identifying a location within the United States of America.
- `directAddress`: MUST contain a Direct Address as defined in the [Applicability Statement](#), formatted as an `addr-spec` as per [RFC 5322](#).

The `telephone` and `directAddress` parameters MAY be multivalued. Multiple values SHALL be expressed as a comma-separated list of valid `patient-parameter-values`. Any white space immediately adjacent to a comma separator SHALL be ignored.

Example:

```
patient: givenName=John; surname=Doe; dateOfBirth=1961-12-31
```

This metadata is included to facilitate patient matching by the recipient when the recipient's patient identifier is not known to the sender. The recipient MAY disregard any characters in the parameter values that are classified under the "MAY" clauses above, e.g., parentheses in a telephone number. If a `patient-id-element` is included for the recipient's domain, the recipient SHOULD disregard the `patient-data-element`.

3.7 Encapsulation of Other Data Types

```
encapsulation-element = "encapsulation:" encapsulated-  
message-type  
encapsulated-message-type = "http" / "hl7v2"
```

This metadata element MUST be included when Direct is used as a transport to encapsulate other message types such as an HTTP request or response, or HL7 v2 message or acknowledgement.

The `encapsulated-message-type "http"` MUST be used for encapsulated HTTP RESTful transactions such as with HL7 Fast Healthcare Interoperability Resources (FHIR) or encapsulated HTTP SOAP transactions such as transactions defined by the Integrating the Healthcare Enterprise (IHE) IT Infrastructure Technical Framework. The `encapsulated-message-type "hl7v2"` MUST be used for encapsulated HL7 V2 transactions.

To encapsulate an HTTP transaction or its response, the HTTP headers and, if applicable, any POST data or other content SHALL be included in one or more separate attachments with the MIME type of `application/x-direct-encapsulated+http`. To encapsulate an HL7 V2 message or its response, the HL7 V2 message data SHALL be included in one or more separate attachments with MIME type of `application/x-direct-encapsulated+hl7v2`. Encapsulated HL7 V2 messages SHALL NOT include any control codes required by the HL7 V2 low level protocol (LLP).

When more than one encapsulated message is included by the sender, the recipient **MUST** process the attachments in the order that the corresponding MIME attachments were included by the sender. A recipient **MUST** process each encapsulated message even if an earlier message results in an error, and **MUST** include the encapsulated responses in the same order in the response message. Thus, the sender **MUST NOT** assemble a sequence of attachments where the message encapsulated in one attachment depends upon or assumes the successful processing of an earlier attachment in the sequence. If the assembled responses would exceed a sender's outgoing message size limits, the sender **MAY** send a failure message instead.

This guide does not provide a mechanism to insert the results of one encapsulated transaction into the next encapsulated transaction when multiple transactions are included in a single message.

4.0 Recipient Responses: Error Conditions

4.1 Version Not Supported

A conforming system **MAY** reject a message when the metadata `version-identifier` specified by the sender is not supported by returning a Delivery Status Notification (DSN) with `action-value` of `failed` and `status-code` of "5.3.3" (indicating "System not capable of selected features").

4.2 Transaction Type Not Supported

A conforming system **MAY** reject a message when the transaction type specified by the sender is not supported by the receiving endpoint by returning a Delivery Status Notification (DSN) with `action-value` of `failed` and `status-code` of "5.3.3" (indicating "System not capable of selected features"). For example, an immunization registry endpoint **MAY** reject a message when the context includes a `type-element` of "type: radiology/order".

4.3 Errors Consuming Content

When an error is encountered by a receiving system after the message has exited the Direct transport layer, the transmission of a DSN or MDN to the sender may no longer be an appropriate or reliable mechanism to communicate to the sender that an error has occurred. For example, in the scenario where the sender requested final delivery notification and the receiving STA has already transmitted a Dispatched MDN back to the sender upon successful delivery of a message to the receiving edge system, section 2.3 of the [Implementation Guide for Delivery Notification in Direct](#) explicitly prohibits the sending STA from notifying the sending system of any subsequently received failure DSN or MDN relating to that message. Thus, in this scenario, if a receiving STA or edge system attempts to signal a subsequent failure by transmitting a DSN or MDN, the sending edge system will not receive these notifications.

A conforming receiving system MAY signal to a sender that the transmitted payload could not be consumed by the receiving system by sending a Direct message to the original sender that includes at least the following context metadata:

- a `type-element` of “`type: error/notification`”, and
- any other metadata elements required to be included by this guide.

The Health Content Container MUST be structured as a multipart/mixed MIME body and MUST NOT be structured as an MDN or DSN. The first part of the Health Content Container MUST be a valid MIME body of type “`text/plain`” containing a description of the problem in a human-readable format. The system that sends the error message MAY request final delivery notification for the error notification. Since the error message is a Direct message, the STA that receives the error message MUST transmit any required MDNs and/or DSNs in accordance with the [Applicability Statement](#) and, when applicable, the [Implementation Guide for Delivery Notification in Direct](#).

Note that this error reporting framework provides for a generalized error reporting mechanism in between the transport and application layers. When an application layer protocol defines its own error reporting mechanism (e.g. HL7 V2 negative acknowledgement messages), the application layer protocol SHOULD be used to signal errors where possible.

The receiving system MAY send an error message containing the above context information even if the error relates to a received message that did not contain context metadata. Note that the system receiving such an error message may not be able to interpret the context metadata if it does not implement this guide.

5.0 Security Considerations

In addition to the security considerations described in the [Applicability Statement](#), implementers should consider security and privacy issues related to patient matching accuracy based on the available metadata.

6.0 Acknowledgements

This document was written by the [Direct Project Implementers Workgroup](#).

Editors:

Luis Maas, EMR Direct
John Hall, Krysora

7.0 References

1. [Applicability Statement for Secure Health Transport v1.2](#)
2. [Implementation Guide for Delivery Notification in Direct v1.0](#)
3. [RFC 822](#) - Standard for the Format of ARPA Internet Text Messages
4. [RFC 2045](#) - Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies
5. [RFC 2046](#) - Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types
6. [RFC 2119](#) - Keywords to use in RFC's for Requirement Levels
7. [RFC 4122](#) - A Universally Unique IDentifier (UUID) URN Namespace
8. [RFC 5234](#) - Augmented BNF for Syntax Specifications: ABNF
9. [E.164](#) - The international public telecommunication numbering plan – ITU
10. [ISO 3166](#) - Country codes and codes for subdivisions

Appendix A Examples

This appendix includes non-normative examples of messages expressing Context. Each example lists a sample Health Content Container conforming to the requirements of the [Applicability Statement](#), prior to message wrapping, signing and encryption.

A.1 Context for a PDF document attachment

In this example, the Context information is used by the sender to identify the attached PDF file as a radiology report and provide the recipient with information about the patient to whom the report pertains. The Context attachment is encoded using the quoted-printable encoding to break up long lines. Note that the equals sign "=" is represented as "=3D" in this encoding.

```
Date: Wed, 31 May 2018 18:32:15 -0700 (PDT)
From: test@direct.phimail-dev.com
To: another@direct.example.com
Message-ID: <0000015c-6148-1d24-9687-50a0730f8b21.test@direct.phimail-dev.com>
Subject: Context Example 1
MIME-Version: 1.0
Content-Type: multipart/mixed;
    boundary="-----_Part_14_125690771.1496280735009"
X-Direct-Context: <0000015c-6148-1bc5-960f-cf885d5b8df1@direct.phimail-
dev.com>

-----_Part_14_125690771.1496280735009
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: quoted-printable

This is the main message content. A PDF radiology report is attached.
-----_Part_14_125690771.1496280735009
Content-Type: text/plain; charset="us-ascii"
Content-Transfer-Encoding: quoted-printable
Content-ID: <0000015c-6148-1bc5-960f-cf885d5b8df1@direct.phimail-dev.com>
Content-Disposition: attachment; filename=metadata.txt

version: 1.1
id: 2ba8a9a1-0f59-4688-b818-67930ae26979
patient-id: 2.16.840.1.113883.19.999999:123456
type: radiology/report
patient: givenName=3DJohn; middleName=3DJacob; surname=3DDoe; dateOfBirth=
=3D1961-12-31; gender=3DM; postalCode=3D12345

-----_Part_14_125690771.1496280735009
Content-Type: application/pdf
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename="report.pdf"

JVBERi0xLjUNCiW1tbW1dQoxIDAgb2JqDQo8PC9UeXB1L0NhdGFsb2cvUGFnZXMGMiAwIFIvTGfu
Zyhlbi1VUykgL1N0cnVjdFRyZWVsb290IDggMCBSL01hcmtJbWZvPDwvTWYya2VkIHRydWU+Pj4+
[....bulk of Base64 encoded PDF file redacted for brevity....]
Pj4NCnN0YXJ0eHJlZg0KMTQ3MDC4DQo1JUVPRg==
-----_Part_14_125690771.1496280735009--
```

A.2 Context for an encapsulated HL7 message

In this example, the payload is an encapsulated HL7 ADT-A01 message:

```
Date: Tue, 30 May 2018 13:37:26 -0700 (PDT)
From: queries@test.phimail-dev.com
To: hl7endpoint@test.directproject.net
Message-ID: <0000015c-5b13-d424-f67e-720eed4b51f1.queries@test.phimail-dev.com>
Subject: Context Example 2
MIME-Version: 1.0
Content-Type: multipart/mixed;
    boundary="-----_Part_0_14860469.1496176645154"
X-Direct-Context: <0000015c-5b13-d424-a351-ed4a58a252ef@test.phimail-dev.com>>

-----_Part_0_14860469.1496176645154
Content-Type: text/plain
Content-ID: <0000015c-5b13-d424-a351-ed4a58a252ef@test.phimail-dev.com>>
Content-Disposition: attachment; filename=metadata.txt

version: 1.1
id: 2142848
encapsulation: hl7v2

-----_Part_0_14860469.1496176645154
Content-Type: application/x-direct-encapsulated+hl7v2
Content-Transfer-Encoding: quoted-printable
Content-Disposition: attachment; filename="adt-sample.hl7"

MSH|^~\&|ADTMON|EMRDIRECT|ADTHUB|PH1|20170530133715||ADT^A01|2142848|T|2.3|=
[.....remainder of HL7 message redacted for brevity.....]
-----_Part_0_14860469.1496176645154--
```

A.3 Error notification message

In this example, the Context information is used to signal an error as per [Section 4.3](#):

```
Date: Tue, 17 June 2018 09:37:26 -0700 (PDT)
From: another@direct.example.com
To: test@direct.phimail-dev.com
Message-ID: <0000015f-53e3-1224-f67e-720eed4b51f3.another@test.phimail-dev.com>
Subject: Error Notification
MIME-Version: 1.0
Content-Type: multipart/mixed;
    boundary="-----_Part_0_15660469.1496176645159"
X-Direct-Context: <0000015f-53e3-1224-f67e-720eed4b51f4@test.phimail-dev.com>

-----_Part_0_15660469.1496176645159
Content-Type: text/plain

The attachment could not be parsed.

-----_Part_0_15660469.1496176645159
Content-Type: text/plain
Content-ID: <0000015f-53e3-1224-f67e-720eed4b51f4@test.phimail-dev.com>
Content-Disposition: attachment; filename=metadata.txt
```

version: 1.1
id: 84848-example-112
type: error/notification

-----=_Part_0_15660469.1496176645159--