

6. Information Objects

The information objects that users exchange in EDI messaging are of two kinds: EDI messages (abEDIMxe "ab:EDIM"§), and EDI notifications (abEDINxe "ab:EDIN"§).

NOTE - The glEDI messaging system userxe "gl:EDI messaging system user"§ (abEDIMG userxe "ab:EDIMG user"§) is normally an glEDI applicationxe "gl:EDI application"§ or computer process, not a person. For brevity, the term gluserxe "gl:user"§ is used throughout the rest of this Recommendation with the meaning of EDIMG user.

```
tyInformationObjectxe "ty:InformationObject"§ ::= CHOICE {  
    edim                [0] EDIM,  
    edin                [1] EDIN }
```

7. Common data types

Information items of several kinds appears both in EDI messages and EDI notifications. These common items are defined below.

7.1 EDIM Identifier

An glEDIM Identifierxe "gl:EDIM Identifier"§ is an information item that unambiguously, globally and forever uniquely identifies an EDIM.

It comprises an OR Name and a string which may for example contain a time or sequence number or other sufficient information to make this EDIM unique.

```
tyEDIMIdentifierxe "ty:EDIMIdentifier"§ ::= SET {  
    user                [0] ORName,  
    user-relative-identifier [1] LocalReference }
```

NOTE - OR Name is defined in 8.5.5 of Recommendation X.411.

The EDIM Identifier shares the same value set with the IPM Identifier defined in Recommendation X.420. Therefore an EDI user agent or EDI message store that is capable of handling both IPM and EDIM shall make sure that the Local Reference is unique both for IPMs and EDIMs.

An EDIM Identifier has the following components:

- a) otUserxe "ot:User"§: Identifies the user who originates the EDIM. One of the user's OR Names.
- b) otUser-relative-identifierxe "ot:User-relative-identifier"§: Unambiguously identifies the EDIM, distinguishing it from all other EDIMs that the user who is identified by the User component originates. A Printable String of from zero to a prescribed number of characters (see annex G). A length of zero is discouraged.

```
tyLocalReference "ty:LocalReference" $ ::= PrintableString (SIZE (0..ub-local-reference))
```

7.2 Extensions

A mechanism is provided which allows for future extensions to this Recommendation.

```
tyExtensionField "ty:ExtensionField" $ ::= SEQUENCE {
  type          [0] EDIM-EXTENSION,
  criticality    [1] Criticality DEFAULT FALSE,
  value         [2] ANY DEFINED BY type DEFAULT NULL NULL }
```

An Extension field can be marked critical (Criticality set to TRUE) or non-critical (Criticality set to FALSE) for acceptance of Responsibility. An extension marked as non-critical for Responsibility may be ignored or discarded, while an extension marked as critical must be known and performed for acceptance of Responsibility of an EDIM.

NOTE - The term glEDIM Responsibility "gl:EDIM Responsibility" \$ is defined in 3.5 of Recommendation F.435. Throughout this document, the term "glResponsibility" "gl:Responsibility" \$ refers to the term defined in Recommendation F.435, and not to the everyday use of the word.

```
tyCriticality "ty:Criticality" $ ::= BOOLEAN
```

As a notation support for future definitions of extensions, a MACRO is defined.

```
maEDIM-EXTENSION "ma:EDIM-EXTENSION" $ MACRO ::=
  BEGIN

    TYPE NOTATION          ::= DataType Critical | empty
    VALUE NOTATION         ::= value(VALUE OBJECT IDENTIFIER)

    DataType               ::= type (X) Default | empty
    Default                 ::= "DEFAULT" value (X) | empty
    Critical                ::= "CRITICAL" | empty

  END -- of extension
```

8. EDI Messages

An glEDI Messagex "gl:EDI Message" \$ (abEDIMx "ab:EDIM" \$) is a member of the primary class of information objects conveyed between users in EDI Messaging.

NOTE 1 - The term glmessagex "gl:message" \$ when used throughout the rest of this Recommendation is a synonym for EDI Message.

```
tyEDIM "ty:EDIM" $ ::= SEQUENCE {
  heading    Heading,
  body       Body }
```

An EDI Message consists of the following components:

- a) Heading: a set of Heading Fields (or Fields), each an information item that gives a characteristic of the EDI Message.

b) Body: a sequence of one or more body parts

```
tyBodyxe "ty:Body"$ ::= SEQUENCE {  
    primary-body-part          PrimaryBodyPart,  
    additional-body-parts      OtherBodyParts OPTIONAL }
```

```
tyPrimaryBodyPartxe "ty:PrimaryBodyPart"$ ::= CHOICE {  
    edi-body-part              [0] EDIBodyPart,  
    forwarded-EDIM             [1] EDIMBodyPart }
```

```
tyOtherBodyPartsx "ty:OtherBodyParts"$ ::= SEQUENCE OF EDIM-ExternallyDefinedBodyPart
```

NOTE 2 - EDIM-Externally Defined Body Part is defined in .

The Body has one Primary Body Part that contains an EDI information object. This body part is either an EDI interchange itself or a forwarded EDIM. Examples of types of EDI information objects are glEDI Interchangesxe "gl:EDI Interchanges"\$ defined by ISO 9735, Electronic data interchange for administration, commerce and transport (abEDIFACTxe "ab:EDIFACT"\$), by United Nations Trade Data Interchange (abUNTDXe "ab:UNTDI"\$) and by American National Standards Institute committee X12 (abANSIX12xe "ab:ANSIX12"\$).

NOTE 3 - The scope of an EDI information object type is rather large and includes for example Privately Defined types. For brevity, the term glinterchangexe "gl:interchange"\$ is used throughout the rest of this Recommendation with the meaning of EDI Interchange.

The following rules comply with the requirements stated in 7.4 of Recommendation F.435:

- c) When an EDIM is first created, the Primary Body Part shall contain one EDI Body Part.
- d) When an EDIM is forwarded, its structure shall comply with the rules given in .

The Primary Body Part has one of two basic forms:

- the Primary Body Part contains an EDI information object
- the Primary Body Part contains a forwarded EDI Message.

Additionally, other body parts may be present in a message related to the Primary Body Part but of a different type. Examples of related body parts might be textual information, voice annotation or graphics to be used in conjunction with the interchange.

The structure of an EDI Message is depicted in Figure .

Error! Cannot open file.§

FIGURE

EDI Message Structureetc "EDI Message Structure" \f F \l 2§

8.1 Heading Field Component Types

Information items of several kinds appear throughout the Heading. These common items are defined below.

In the text that follows, reference is made to EDIFACT segments and data elements. Annex K to this Recommendation explains this in relation to UNTDI and ANSI X12. Values copied from EDI data elements and represented as T.61 Strings are semantically equivalent to the characters used to form the EDI data elements in EDIFACT, UNTDI and ANSI X12.

8.1.1 Interchange Recipient / Sender

The Interchange Recipient and Interchange Sender fields have some data types in common. They are defined below.

8.1.1.1 Identification Code

The Identification Code identifies a sender/recipient of an interchange. This is semantically identical to the "Sender identification/recipient identification code" component of the Interchange sender/recipient of the EDIFACT UNB segment.

`tyIdentificationCode`xe "ty:IdentificationCode"§ ::= T61String (SIZE (1..ub-identification-code))

8.1.1.2 Identification Code Qualifier

The Identification Code Qualifier, if present, is a qualifier to the Identification Code of a sender/recipient. This is semantically identical to the "Identification code qualifier" component of the Interchange sender/recipient of the EDIFACT UNB segment.

`tyIdentificationCodeQualifier`xe "ty:IdentificationCodeQualifier"§ ::= T61String (SIZE (1..ub-identification-code-qualifier))

8.1.1.3 Routing Address

The Routing Address, if present, is an address for routing to the sender/recipient specified in the Identification Code. This is semantically identical to the "Address for reverse routing / Routing address" component of the Interchange sender/recipient of the EDIFACT UNB segment.

`tyRoutingAddress`xe "ty:RoutingAddress"§ ::= T61String (SIZE (1..ub-routing-address))

8.2 Heading Fields

The fields that may appear in the Heading of an EDIM are defined and described below.

```
tyHeadingxe "ty:Heading"§ ::= SEQUENCE {  
    this-EDIM [1] ThisEDIMField,  
    originator [2] OriginatorField OPTIONAL,  
    recipients [3] RecipientsField OPTIONAL,  
    edin-receiver [4] EDINReceiverField OPTIONAL,  
    responsibility-forwarded [5] ResponsibilityForwarded DEFAULT FALSE,  
    edi-bodypart-type [6] EDIBodypartType DEFAULT {id-edifact-ISO646},  
    incomplete-copy [7] IncompleteCopyField DEFAULT FALSE,  
    expiry-time [8] ExpiryTimeField OPTIONAL,  
    related-messages [9] RelatedMessagesField OPTIONAL,  
    obsoleted-EDIMs [10] ObsoletedEDIMsField OPTIONAL,  
    edi-application-security-elements [11] EDIApplicationSecurityElementsField OPTIONAL,  
    cross-referencing-information [12] CrossReferencingInformationField OPTIONAL,  
  
    -- Begin Fields from EDIFACT Interchange  
    edi-message-type [13] EDIMessageTypeField OPTIONAL,  
    service-string-advice [14] ServiceStringAdviceField OPTIONAL,  
    syntax-identifier [15] SyntaxIdentifierField OPTIONAL,  
    interchange-sender [16] InterchangeSenderField OPTIONAL,  
    date-and-time-of-preparation [17] DateAndTimeOfPreparationField OPTIONAL,  
    application-reference [18] ApplicationReferenceField OPTIONAL,  
    -- End Fields from EDIFACT  
  
    heading-extensions [19] HeadingExtensionsField OPTIONAL }
```

NOTE - The names of the Heading fields derived from EDI standards are taken directly from the relevant standards. See also Annex K.

8.2.1 This EDIM

The This EDIM field identifies the EDIM. It comprises an EDIM Identifier which provides a globally and forever unique identification for the EDIM.

```
tyThisEDIMFieldxe "ty:ThisEDIMField"§ ::= EDIMIdentifier
```

NOTE - EDIM Identifier is defined in .

8.2.2 Originator

Identifies the EDIM's originator. It comprises an OR Name. If the Originator field is not present in the EDIM Heading on reception, then the Originating-name of the delivery envelope shall be used to determine the originator of the EDIM (see 8.2.1.1.1.1 of Recommendation X.411).

```
tyOriginatorFieldxe "ty:OriginatorField"§ ::= ORName
```

NOTE - OR Name is defined in 8.5.5 of Recommendation X.411.

8.2.3 Recipients

The Recipients field identifies the user(s) and distribution lists(abDLxe "ab:DL"§) who are the (preferred) recipient(s) of the EDIM. It comprises a set of Recipients subfields, one for each recipient. If the Recipients field is not present in the EDIM Heading on reception, then the This-recipient-name of the delivery envelope shall be used to determine the recipient of the EDIM (see 8.3.1.1.1.3 of Recommendation X.411).

NOTE - The fact that a message can be redirected or forwarded is reflected in the word "preferred" above.

tyRecipientsFieldxe "ty:RecipientsField"§ ::= SET OF RecipientsSubField

The Recipients subfield is an information item that identifies a recipient of an EDIM and that may make certain requests of him.

```
tyRecipientsSubFieldxe "ty:RecipientsSubField"§ ::= SEQUENCE {
  recipient                                [1] RecipientField,
  action-request                          [2] ActionRequestField DEFAULT {id-for-action},
  edi-notification-requests-field         [3] EDINotificationRequestsField OPTIONAL,
  responsibility-passing-allowed           [4] ResponsibilityPassingAllowedField DEFAULT FALSE,

  -- Begin Fields from EDIFACT UNB
  interchange-recipient                   [5] InterchangeRecipientField OPTIONAL,
  recipient-reference                     [6] RecipientReferenceField OPTIONAL,
  interchange-control-reference            [7] InterchangeControlReferenceField OPTIONAL,
  processing-priority-code                 [8] ProcessingPriorityCodeField OPTIONAL,
  acknowledgement-request                 [9] AcknowledgementRequestField DEFAULT FALSE,
  communications-agreement-id             [10] CommunicationsAgreementIdField OPTIONAL,
  test-indicator                          [11] TestIndicatorField DEFAULT FALSE,
  -- End Fields from EDIFACT UNB

  -- Begin Fields from ANSI X12 ISA
  authorization-information                [12] AuthorizationInformationField OPTIONAL,
  -- End Fields from ANSI X12 ISA

  recipient-extensions                    [13] RecipientExtensionsField OPTIONAL }

```

The Recipients subfield has the following components:

8.2.3.1 Recipient

A otRecipientxe "ot:Recipient"§ identifies the preferred recipient in question. It comprises an OR Name.

tyRecipientFieldxe "ty:RecipientField"§ ::= ORName

NOTE - OR Name is defined in 8.5.5 of Recommendation X.411.

8.2.3.2 Action Request

An otAction Requestxe "ot:Action Request"§ indicates what action the originator requests from the recipient. Its value is an object identifier.

tyActionRequestFieldxe "ty:ActionRequestField"§ ::= OBJECT IDENTIFIER

The following standard values have object identifiers defined in this Recommendation:

- For Action
- Copy

The absence of this field shall be interpreted as having the default value set to For Action.

NOTE - Additional values for this field can be defined by any interested parties.

8.2.3.3 *EDI Notification Requests*

The otEDI Notification Requestsxe "ot:EDI Notification Requests"\$ component (Default: no notifications, no notification security and no reception security) may make certain requests of the preferred recipient denoted by the Recipient field.

NOTE - The fact that a message can be redirected or forwarded is reflected in the word "preferred" above.

```
tyEDINotificationRequestsFieldxe "ty:EDINotificationRequestsField"$ ::= SEQUENCE {  
    edi-notification-requests          [0] EDINotificationRequests DEFAULT {},  
    edi-notification-security          [1] EDINotificationSecurity DEFAULT {},  
    edi-reception-security              [2] EDIReceptionSecurity DEFAULT {} }
```

```
tyEDINotificationRequestsxe "ty:EDINotificationRequests"$ ::= BIT STRING {  
    pn (0),  
    nn (1),  
    fn (2) }(SIZE (0..ub-bit-options))
```

```
tyEDINotificationSecurityxe "ty:EDINotificationSecurity"$ ::= BIT STRING {  
    proof (0),  
    non-repudiation (1) }(SIZE (0..ub-bit-options))
```

```
tyEDIReceptionSecurityxe "ty:EDIReceptionSecurity"$ ::= BIT STRING {  
    proof (0),  
    non-repudiation (1) }(SIZE (0..ub-bit-options))
```

The EDI Notification Requests field consists of a sequence of three bit strings of which the first selects the type of notification, the second selects what security function should be applied to that notification, and the third may make certain security requests for proof or non-repudiation of reception of this EDIM by the recipient. EDI Notification Security and EDI Reception Security shall not be requested if EDI Notifications are not requested.

The EDI Notification Requests bit string may assume any of the following values simultaneously:

- a) PN: A notification of acceptance of Responsibility is requested in the circumstances prescribed in .

- b) NN: A notification of refusal of Responsibility of a message is requested in the circumstances prescribed in .
- c) FN: A forwarded notification is requested in the circumstances prescribed in .

The absence of the EDI Notification Requests bit string implies that no EDI Notification requests are made.

The EDI Notification Security bit string may assume any of the following values simultaneously:

- d) proof: When submitting the EDIN to the MTS, content-integrity-check shall be requested in the Message-submission-argument as defined in 8.2.1.1.1.28 in Recommendation X.411.
- e) non-repudiation: When submitting the EDIN to the MTS, content-integrity-check shall be requested in the Message-submission-argument as defined in 8.2.1.1.1.28 in Recommendation X.411 with a non-repudiable certificate.

The absence of the EDI Notification Security bit string implies that no EDI Notification Security requests are made.

The EDI Reception Security bit string may assume any of the following values simultaneously:

- f) proof: When submitting the EDIN to the MTS, content-integrity-check (possibly in the message token), or the message-origin-authentication-check (depending on the security policy in force) shall be requested. A notification shall contain the security elements and shall be signed on submission to the MTS, using content-integrity-check (possibly in the message token) or message-origin-authentication-check (depending on the security policy in force) in the Message-submission-argument as defined in 8.2.1.1.1.26, 28 and 29 of Recommendation X.411.
- g) non-repudiation: When submitting the EDIN to the MTS, a non-repudiable content-integrity-check (possibly in the message token) or a message-origin-authentication-check (depending on the security policy in force) shall be requested. A notification shall contain the security elements and shall be signed on submission to the MTS, using non-repudiable content-integrity-check (possibly in the message token) or message-origin-authentication-check (depending on the security policy in force) in the Message-submission-argument as defined in 8.2.1.1.1.26, 28 and 29 of Recommendation X.411 .

The absence of the EDI Reception Security field implies that no EDI Reception Security requests are made.

NOTE - Security services are available only if the MTA supports secure messaging.

8.2.3.4 *Responsibility Passing Allowed*

The Responsibility Passing Allowed field indicates that forwarding Responsibility is allowed if this field is set to TRUE. Absence of the field shall be interpreted as the value FALSE.

A recipient of a message with the Responsibility Passing Allowed field set to FALSE shall originate EDIN's as requested, and shall not forward Responsibility.

```
tyResponsibilityPassingAllowedFieldxe "ty:ResponsibilityPassingAllowedField"$ ::= BOOLEAN -- Default FALSE
```

If allowed, Responsibility may be forwarded to at most one recipient.

8.2.3.5 *Interchange Recipient*

The otInterchange Recipientxe "ot:Interchange Recipient"\$ identifies the EDI Interchange recipient. This is semantically identical to the "Interchange recipient" of the EDIFACT UNB segment.

```
tyInterchangeRecipientFieldxe "ty:InterchangeRecipientField"$ ::= SEQUENCE {  
  recipient-identification-code      [0] IdentificationCode,  
  identification-code-qualifier      [1] IdentificationCodeQualifier OPTIONAL,  
  routing-address                    [2] RoutingAddress OPTIONAL }
```

NOTE - The above fields are defined in .

8.2.3.6 *Recipient Reference*

The otRecipient Referencexe "ot:Recipient Reference"\$ identifies a reference meaningful to the recipient's EDI application. This is semantically identical to the "Recipient's Reference, Password" of the EDIFACT UNB segment. It consists of two strings.

```
tyRecipientReferenceFieldxe "ty:RecipientReferenceField"$ ::= SEQUENCE {  
  recipient-reference                [0] RecipientReference,  
  recipient-reference-qualifier      [1] RecipientReferenceQualifier OPTIONAL }
```

```
tyRecipientReferencexe "ty:RecipientReference"$ ::= T61String (SIZE (1..ub-recipient-reference))
```

```
tyRecipientReferenceQualifierxe "ty:RecipientReferenceQualifier"$ ::= T61String (SIZE (1..ub-recipient-reference-qualifier))
```

8.2.3.7 *Interchange Control Reference*

Indicates the Interchange Control Reference as assigned by the Interchange sender. This is semantically identical to the "Interchange control reference" of the EDIFACT UNB segment.

```
tyInterchangeControlReferenceFieldxe "ty:InterchangeControlReferenceField"$ ::= T61String (SIZE (1..ub-interchange-control-reference))
```

8.2.3.8 *Processing Priority Code*

Indicates the EDI application Processing Priority Code. This is semantically identical to the "Processing priority code" in the EDIFACT UNB segment. It consists of a string.

```
tyProcessingPriorityCodeFieldxe "ty:ProcessingPriorityCodeField"$ ::= T61String (SIZE (1..ub-processing-  
priority-code))
```

8.2.3.9 *Acknowledgement Request*

The Acknowledgement Request indicates the request for EDI acknowledgement as indicated by the interchange sender. This is semantically identical to the "Acknowledgement request" in the EDIFACT UNB segment. Its value is a Boolean, where the value TRUE indicates a request for acknowledgement. Absence of this field shall be interpreted as the value FALSE.

```
tyAcknowledgementRequestFieldxe "ty:AcknowledgementRequestField"$ ::= BOOLEAN -- default FALSE
```

8.2.3.10 *Communications Agreement Id*

The Communications Agreement Id indicates the type of Communications Agreement controlling the interchange, e.g. Customs or other agreement. This is semantically identical to the "Communications agreement id" in the EDIFACT UNB segment.

```
tyCommunicationsAgreementIdFieldxe "ty:CommunicationsAgreementIdField"$ ::= T61String (SIZE (1..ub-  
communications-agreement-id))
```

8.2.3.11 *Test Indicator*

Indicates that the EDI Interchange is a test. This is semantically identical to the "Test indicator" in the EDIFACT UNB segment. It is a Boolean where the value TRUE indicates that the EDI Interchange is a test. Absence of this field shall be interpreted as the value FALSE.

```
tyTestIndicatorFieldxe "ty:TestIndicatorField"$ ::= BOOLEAN -- default FALSE
```

8.2.3.12 *Authorization Information*

The Authorization Information indicates who authorized the interchange. This is semantically identical to the "Authorization information" in the ANSI X12 Interchange.

```
tyAuthorizationInformationFieldxe "ty:AuthorizationInformationField"$ ::= SEQUENCE {  
    authorization-information [0] AuthorizationInformation,  
    authorization-information-qualifier [1] AuthorizationInformationQualifier OPTIONAL }
```

```
tyAuthorizationInformationxe "ty:AuthorizationInformation"$ ::= T61String (SIZE (1..ub-authorization-  
information))
```

tyAuthorizationInformationQualifierxe "ty:AuthorizationInformationQualifier"\$::= T61String (SIZE (1..ub-authorization-information-qualifier))

NOTE - In the above text reference is made to ANSI X12 segments and data elements. Annex K to this Recommendation explains this in relation to UNTDI and EDIFACT (ISO 9735), being the other two widely used syntaxes.

8.2.3.13 Recipient Extensions

The Recipient otExtensionsxe "ot:Extensions"\$ contains extensions to the Recipients subfield.

tyRecipientExtensionsFieldxe "ty:RecipientExtensionsField"\$::= SET OF RecipientExtensionsSubField

tyRecipientExtensionsSubFieldxe "ty:RecipientExtensionsSubField"\$::= ExtensionField

There are no extensions defined in this Recommendation.

8.2.4 EDIN Receiver

Identifies the recipient to whom EDINs are to be sent. This is created by the originator of the EDIM when the Recipient of a requested notification is different from the Originator of the message. It consists of a sequence of OR Name, EDIM Identifier and First Recipient.

This field shall not be present if EDI Notification Requests are not made.

This field shall be present in a forwarded message when the forwarding EDI user agent; (abEDI-UAxe "ab:EDI-UA"\$) or EDI message store; (abEDI-MSxe "ab:EDI-MS"\$) forwards Responsibility. This field may be present when the forwarding EDI-UA accepts Responsibility. Rules related to the construction of this field are given in .

NOTE 1 - For brevity, the term user agent (abUAxe "ab:UA"\$) is used throughout the rest of this Recommendation with the meaning of EDI-UA, and the term message store (abMSxe "ab:MS"\$) is used throughout the rest of this Recommendation with the meaning of EDI-MS.

tyEDINReceiverFieldxe "ty:EDINReceiverField"\$::= SEQUENCE {	
edin-receiver-name	[0] ORName,
original-edim-identifier	[1] EDIMIdentifier OPTIONAL,
first-recipient	[2] FirstRecipientField OPTIONAL}

The "first-recipient" field shall not be present if more than one recipient contains EDI Notification Requests.

The "original-edim-identifier" and the "first-recipient" fields shall not be present when the Primary Body Part is an EDI Body Part (that is, when the original originator first creates the EDIM).

NOTE 2 - The Original EDIM Identifier and First Recipient fields are included in

order to allow the recipient to construct the EDIN for a forwarded EDIM. See (more specifically) and for rules related to the construction of an EDIN; see for rules related to the First Recipient field when constructing a forwarded EDIM. OR Name is defined in 8.5.5 of Recommendation X.411. First Recipient Field is defined in .

8.2.5 Responsibility Forwarded

The Responsibility Forwarded field is used to indicate whether Responsibility was forwarded. Absence of this field shall be interpreted as the value FALSE.

tyResponsibilityForwarded ::= BOOLEAN -- Default False

If this field has the value TRUE it indicates to a receiving UA that Responsibility was forwarded. If this field has the value FALSE (or is absent) it indicates to a receiving UA that the security elements of the inner envelope have been checked.

Subject to the security policy in force, the security elements may have been checked when the message was forwarded. However, when Responsibility is accepted, the security elements shall be checked.

NOTE - Rules regarding the use of this field are contained in and .

8.2.6 EDI Body Part Type

Indicates the EDI standard and EDI character sets used in the Primary Body Part. It is represented by a single object identifier.

NOTE 1 - This object identifier implements the element of service Character Set defined in Recommendation F.435.

tyEDIBodyPartType ::= OBJECT IDENTIFIER -- default EDIFACT-ISO646

The following standard values have object identifiers defined in this Recommendation:

- EDIFACT: ISO646|T61|UNDEFINED OCTETS
- ANSIX12: ISO646|T61|EBCDIC|UNDEFINED OCTETS
- UNTDI: ISO646|T61|UNDEFINED OCTETS
- PRIVATE: UNDEFINED OCTETS
- UNDEFINED: UNDEFINED OCTETS

The absence of this field shall be interpreted as having the default value set to EDIFACT, ISO 646.

NOTE 2 - The character set referred to by the object identifier is that in which both the EDI Body Part, and those Heading fields that are OCTET STRINGS and are derived from the EDI Interchange are encoded, notwithstanding the fact that these types are defined as OCTET STRING.

The value of the EDI Body Part Type field shall be used in the Encoded Information Types (abEITxe "ab:EIT"§) in the MTS abstract operations (in accordance with). This enables a UA to signal to the MTS what type of EDI standard the EDIM's Primary Body Part complies with. The MTS shall make use of this information, if the recipient UA has registered delivery restrictions on Encoded Information Types, to decide if it can deliver the EDIM.

NOTE - The term glEncoded Information Typexe "gl:Encoded Information Type"§ is defined in 8.1 of Recommendation X.402. See also 8.2.1.1.1.23 of Recommendation X.411.

8.2.7 Incomplete Copy

The Incomplete Copy field indicates that the subject EDIM is an incomplete copy of an EDIM. Its value is a Boolean. This field shall have the value TRUE if body parts are removed when an EDIM is forwarded. The absence of this field shall be interpreted as having the value FALSE.

tyIncompleteCopyFieldxe "ty:IncompleteCopyField"§ ::= BOOLEAN -- Default False

NOTE - The term "subject EDIM" is defined in .

8.2.8 Expiry Time

Indicates when the originator considers this EDIM loses its validity. It comprises a date and time (UTC).

tyExpiryTimeFieldxe "ty:ExpiryTimeField"§ ::= UTCTime

8.2.9 Related Messages

Identifies messages that the originator of this EDIM considers related to it. It comprises a sequence of one or more message references, one for each related message.

tyRelatedMessagesFieldxe "ty:RelatedMessagesField"§ ::= SEQUENCE OF RelatedMessageReference

tyRelatedMessageReferencexe "ty:RelatedMessageReference"§ ::= CHOICE {
edi-message-reference [0] EDIMIdentifier,
external-message-reference [1] ExternalMessageReference }

tyExternalMessageReferencexe "ty:ExternalMessageReference"§ ::= EXTERNAL

8.2.10 *Obsoleted EDIMs*

The Obsoleted EDIMs Field identifies one or more EDIMs that the present EDIM obsoletes. It is a sequence of subfields, each an EDIM Identifier.

tyObsoletedEDIMsFieldxe "ty:ObsoletedEDIMsField"\$::= SEQUENCE OF ObsoletedEDIMsSubfield

tyObsoletedEDIMsSubfieldxe "ty:ObsoletedEDIMsSubfield"\$::= EDIMIdentifier

8.2.11 *EDI Application Security Elements*

The EDI Application Security Elements field allows an EDI application to exchange security elements having an end-to-end significance.

tyEDIApplicationSecurityElementsFieldxe "ty:EDIApplicationSecurityElementsField"\$::= SEQUENCE {
 edi-application-security-element [0] EDIApplicationSecurityElement OPTIONAL,
 edi-encrypted-primary-bodypart [1] BOOLEAN OPTIONAL,
 edi-application-security-extensions [2] EDIApplicationSecurityExtensions OPTIONAL }

tyEDIApplicationSecurityElementxe "ty:EDIApplicationSecurityElement"\$::= BIT STRING (SIZE (0..ub-edi-application-security-elements))

tyEDIApplicationSecurityExtensionsxe "ty:EDIApplicationSecurityExtensions"\$::= SEQUENCE OF EDIApplicationSecurityExtension

tyEDIApplicationSecurityExtensionxe "ty:EDIApplicationSecurityExtension"\$::= ExtensionsField

8.2.12 *Cross Referencing Information*

The Cross Referencing Information allows an EDI application to reference individual body parts within the same EDIM and within other EDIMs. It contains a set of cross reference data. Its usage is outside the scope of this Recommendation.

tyCrossReferencingInformationFieldxe "ty:CrossReferencingInformationField"\$::= SET OF CrossReferencingInformationSubField

tyCrossReferencingInformationSubFieldxe "ty:CrossReferencingInformationSubField"\$::= SEQUENCE {
 application-cross-reference [0] ApplicationCrossReference,
 message-reference [1] MessageReference OPTIONAL,
 body-part-reference [2] BodyPartReference }

tyApplicationCrossReferencexe "ty:ApplicationCrossReference"\$::= OCTET STRING

tyMessageReferencexe "ty:MessageReference"\$::= EDIMIdentifier

NOTE - Body Part Reference is defined in .

8.2.13 *EDI Message Type*

Indicates the Message type(s) present in the EDI Interchange. It consists of a set of strings.

NOTE - "Message" is to be understood as message types that are defined in EDI standards and shall not be confused with "message" used elsewhere in this Recommendation.

```
tyEDIMessageTypeFieldxe "ty:EDIMessageTypeField"$ ::= SET OF EDIMessageTypeFieldSubField
```

```
tyEDIMessageTypeFieldSubFieldxe "ty:EDIMessageTypeFieldSubField"$ ::= T61String (SIZE (1..ub-edimessage-type))
```

The values for this field shall be:

- EDIFACT: Message Type from the UNH segment
- ANSIX12: Transaction Set ID from the ST segment
- UNTDI: Message Type from the MHD segment.

8.2.14 *Service String Advice*

Indicates the Service String Advice of the EDI Interchange. This is semantically identical to the "UNA, Service string advice" of the EDIFACT Interchange.

```
tyServiceStringAdviceFieldxe "ty:ServiceStringAdviceField"$ ::= SEQUENCE {
  component-data-element-separator    [0] ComponentDataElementSeparator,
  data-element-separator               [1] DataElementSeparator,
  decimal-notation                     [2] DecimalNotation,
  release-indicator                    [3] ReleaseIndicator OPTIONAL,
  reserved                             [4] Reserved OPTIONAL,
  segment-terminator                   [5] SegmentTerminator }
```

```
tyComponentDataElementSeparatorxe "ty:ComponentDataElementSeparator"$ ::= OCTET STRING
(SIZE (1))
```

```
tyDataElementSeparatorxe "ty:DataElementSeparator"$ ::= OCTET STRING (SIZE (1))
```

```
tyDecimalNotationxe "ty:DecimalNotation"$ ::= OCTET STRING (SIZE (1))
```

```
tyReleaseIndicatorxe "ty:ReleaseIndicator"$ ::= OCTET STRING (SIZE (1))
```

```
tyReservedxe "ty:Reserved"$ ::= OCTET STRING (SIZE (1))
```

```
tySegmentTerminatorxe "ty:SegmentTerminator"$ ::= OCTET STRING (SIZE (1))
```

8.2.15 *Syntax Identifier*

Indicates the syntax used. This is semantically identical to the "Syntax identifier" of the EDIFACT UNB segment.

It consists of a sequence of the Syntax Identifier and the Syntax Version.

```
tySyntaxIdentifierFieldxe "ty:SyntaxIdentifierField"$ ::= SEQUENCE {
  syntax-identifier    SyntaxIdentifier,
  syntax-version       SyntaxVersion }
```

```
tySyntaxIdentifierxε "ty:SyntaxIdentifier"§ ::= T61String (SIZE (1..ub-syntax-identifier))
```

```
tySyntaxVersion = "ty:SyntaxVersion"§ ::= PrintableString (SIZE (1..ub-syntax-version))
```

8.2.16 Interchange Sender

Indicates the sender of the EDI Interchange. This is semantically identical to the "Interchange sender" of the EDIFACT UNB segment.

```
tyInterchangeSenderFieldxe "ty:InterchangeSenderField"$s ::= SEQUENCE {
    sender-identification           [0] IdentificationCode,
    identification-code-qualifier  [1] IdentificationCodeQualifier OPTIONAL,
    address-for-reverse-routing    [2] RoutingAddress OPTIONAL } -- EDIFACT Routing Information
```

NOTE - The above fields are defined in .

8.2.17 Date and Time of Preparation

Indicates the Date/Time of preparation of the EDIM. This is in UTC Time and is derived from the "Date and time of preparation" of the EDIFACT UNB segment. It comprises a UTC time.

```
tyDateAndTimeOfPreparationFieldxe "ty:DateAndTimeOfPreparationField" § ::= UTCTime
```

8.2.18 Application Reference

Provides a general reference to an application or function. This is semantically identical to the "Application reference" segment of the EDIFACT UNB segment. It consists of a string.

```
tyApplicationReferenceFieldx "ty:ApplicationReferenceField" § ::= T61String (SIZE (1..ub-application-reference))
```

8.2.19 *Heading Extensions*

The Heading Extensions allows for future extensions to the Heading.

$$\text{tyHeadingExtensionsField} \text{ "ty:HeadingExtensionsField" } \S ::= \text{SET OF HeadingExtensionsSubField}$$
$$\text{tyHeadingExtensionsSubField} \text{ "ty:HeadingExtensionsSubField" } \S ::= \text{ExtensionField}$$

There is no Extensions to the Heading defined in this Recommendation.

NOTE - The Heading Extensions may be used to implement the element of service "Services Indication" defined in Recommendation F.435.

8.3 Body Part Types

The types of body parts that may appear in the Body of an EDIM are defined and described below.

8.3.1 EDI Body Part

An EDI Body Part carries a single EDI Interchange.

```
tyEDIBodyPartxe "ty:EDIBodyPart"$ ::= OCTET STRING
```

The reference definition of EDI Interchange used is that used by EDIFACT (ISO 9735). Annex K to this Recommendation describes equivalent terms in other EDI standards.

8.3.2 EDIM Body Part

An EDIM Body Part contains an EDIM, and optionally, its delivery envelope. It is used for forwarding of EDIMs. When an EDIM is forwarded, its structure shall comply with the rules given in .

```
tyEDIMBodyPartxe "ty:EDIMBodyPart"$ ::= SEQUENCE {  
  parameters          [0] MessageParameters OPTIONAL,  
  data                [1] MessageData }
```

```
tyMessageParametersxe "ty:MessageParameters"$ ::= SET {  
  delivery-time        [0] MessageDeliveryTime OPTIONAL,  
  delivery-envelope    [1] OtherMessageDeliveryFields OPTIONAL,  
  other-parameters    [2] EDISupplementaryInformation OPTIONAL }
```

```
tyMessageDataxe "ty:MessageData"$ ::= SEQUENCE {  
  heading             Heading,  
  body                BodyOrRemoved }
```

```
tyBodyOrRemovedxe "ty:BodyOrRemoved"$ ::= SEQUENCE {  
  primary-or-removed  PrimaryOrRemoved,  
  additional-body-parts AdditionalBodyParts OPTIONAL }
```

```
tyPrimaryOrRemovedxe "ty:PrimaryOrRemoved"$ ::= CHOICE {  
  removed-edi-body    [0] NULL,  
  primary-body-part   [1] EXPLICIT PrimaryBodyPart }
```

```
tyAdditionalBodyPartsx "ty:AdditionalBodyParts"$ ::= SEQUENCE OF CHOICE {  
  external-body-part  [0] EDIM-ExternallyDefinedBodyPart,  
  place-holder        [1] BodyPartPlaceHolder } -- This type is for Body Part Removal
```

```
tyBodyPartPlaceHolderxe "ty:BodyPartPlaceHolder"$ ::= EDIM-ExternallyDefinedBodyPart -- Only the data  
-- portion of the Externally Defined Body shall be removed.  
-- See text in .
```

```
tyEDISupplementaryInformationxe "ty:EDISupplementaryInformation"$ ::= IA5String (SIZE (1..ub-  
supplementary-info-length))
```

NOTE - Primary Body Part is defined in . Body Part Reference is defined in . Message Delivery Time and Other Message Delivery Fields are defined in 8.3.1.1 of

Recommendation X.411.

The Body Part Place Holder is only used for removal of body parts. It may consist of only the Body Part Reference, or a modified Externally Defined Body Part. In the latter case the Object Identifier and Body Part Reference of the removed body part are preserved; from the parameter (if present) and data portions of the removed body part, only the Object Identifier and the tag of the "encoding" field of the EXTERNAL are preserved, that is, the EXTERNAL type shall have no content.

The delivery envelope shall be present if security services are invoked.

The structure of an EDIM Body Part is depicted in Figure .

Error! Cannot open file.§

FIGURE

EDIM Body Part Structureetc "EDIM Body Part Structure" \f F \l 2§

8.3.3 Externally Defined Body Parts

Additional body parts, that relate to the Primary Body Part, may be carried together with an EDI Body Part. These body parts shall not be or include EDI Interchanges.

Additional body parts are externally defined and represent information objects whose semantics and abstract syntax are denoted by an object identifier which the body part carries. They have Parameters and Data components and optionally a Body Part Reference that may be used for cross-referencing to a body part.

```
tyEDIM-ExternallyDefinedBodyPartxe "ty:EDIM-ExternallyDefinedBodyPart"$ ::= SEQUENCE {  
    body-part-reference [0] BodyPartReference OPTIONAL,  
    external-body-part [1] ExternallyDefinedBodyPart -- from IPMS --}
```

```
tyBodyPartReferencexe "ty:BodyPartReference"$ ::= INTEGER -- shall be unique within a EDIM
```

Body Part Reference is assigned when the body part is created, and is not modified subsequently. It shall be present if the originator wishes to cross-reference the body part at creation or in the future.

NOTE - Some otExternally Definedxe "ot:Externally Defined"\$ body part types are defined in 7.3.12 of Recommendation X.420.

9. EDI Notifications

An glEDI Notificationxe "gl:EDI Notification"\$ (abEDINxe "ab:EDIN"\$) is a member of a secondary class of information object conveyed between users in EDI Messaging.

NOTE - The term glnotificationxe "gl:notification"\$ is used throughout the rest of this Recommendation as a synonym for EDI Notification.

tyEDINxe "ty:EDIN"\$::= CHOICE {	
positive-notification	[0] PositiveNotificationFields,
negative-notification	[1] NegativeNotificationFields,
forwarded-notification	[2] ForwardedNotificationFields }

tyPNxe "ty:PN"\$::= EDIN -- with positive-notification chosen

tyNNxe "ty:NN"\$::= EDIN -- with negative-notification chosen

tyFNxe "ty:FN"\$::= EDIN -- with forwarded-notification chosen

- a) glpositive notificationxe "gl:positive notification"\$ (abPNxe "ab:PN"\$): An EDIN that reports its originator's acceptance of Responsibility of an EDIM.
- b) glnegative notificationxe "gl:negative notification"\$ (abNNxe "ab:NN"\$): An EDIN that reports its originator's refusal to accept Responsibility of an EDIM.
- c) glforwarded notificationxe "gl:forwarded notification"\$ (abFNxe "ab:FN"\$): An EDIN that reports that Responsibility of an EDIM has been forwarded together with the subject EDIM.

The EDIM to which an EDIN refers is called the glsubject EDIMxe "gl:subject EDIM"\$ (see also).

The recipient of the EDIN is the Originator of the subject EDIM, or, if present, the OR Name indicated in the EDIN Receiver field. There shall be at most one recipient specified for an EDIN. There shall be at most one PN, NN or FN originated for each subject EDIM by each recipient of whom notifications are requested (except that an NN may be originated by the same UA subsequent to an FN, in accordance with c) of). One FN is originated, if and only if requested, by each recipient that forwards an EDIM. In accordance with the provisions of , the original originator shall receive at most one PN or NN for each recipient of whom notifications were requested, regardless of how many times the EDIM is forwarded, and may receive multiple FNs.

An EDIN consists of Positive, Negative or Forwarded Notification fields. Each of these contains the Common fields which are described below.

The structure of an EDIN is depicted in Figure .

Error! Cannot open file.§

FIGURE

EDI Notification Structureetc "EDI Notification Structure" \f F \l 2§

9.1 Common Fields

The Common fields are defined and described below.

```

tyCommonFieldsxe "ty:CommonFields"$ ::= SEQUENCE {
    subject-edim                [1] SubjectEDIMField,
    edin-originator             [2] EDINOriginatorField,
    first-recipient             [3] FirstRecipientField OPTIONAL,
    notification-time           [4] NotificationTimeField,
    notification-security-elements [5] SecurityElementsField OPTIONAL,
    edin-initiator              [6] EDINInitiatorField,
    notifications-extensions    [7] NotificationExtensionsField OPTIONAL }

```

NOTE - The Common fields appear in the Positive Notification, Negative Notification and Forwarded Notification fields defined below.

9.1.1 Subject EDIM Identifier

The Subject EDIM Identifier is the EDIM Identifier either passed in the EDIN Receiver field, if Responsibility has been forwarded, or the This EDIM field, if not.

```

tySubjectEDIMFieldxe "ty:SubjectEDIMField"$ ::= EDIMIdentifier

```

NOTE - EDIM Identifier is defined in . Subject EDIM is defined in .

9.1.2 EDI Notification Originator

The EDI Notification Originator contains the OR Name of the UA constructing the notification.

```

tyEDINOriginatorFieldxe "ty:EDINOriginatorField"$ ::= ORName

```

NOTE - OR Name is defined in 8.5.5 of Recommendation X.411.

9.1.3 First Recipient

The glFirst Recipientxe "gl:First Recipient"\$ field contains the OR Name of the first recipient in a forwarding chain. This field, together with other fields, is used by the recipient of the notification to correlate the notification and the original message.

```

tyFirstRecipientFieldxe "ty:FirstRecipientField"$ ::= ORName

```

NOTE - OR Name is defined in 8.5.5 of Recommendation X.411

If the originator of the EDIN is not the recipient specified by the original originator, then the First Recipient Field shall be present in the EDIN (see and more specifically).

9.1.4 Notification Time

Notification Time contains the date and time, in UTC format, at which the notification for the subject EDIM was generated.

```

tyNotificationTimeFieldxe "ty:NotificationTimeField"$ ::= UTCTime

```

9.1.5 Security Elements

The Security Elements field is used to provide "proof/non repudiation of content received", "EDI application security" services.

```
tySecurityElementsFieldxe "ty:SecurityElementsField"§ ::= SEQUENCE {  
    original-content                [0] Content OPTIONAL,  
    original-content-integrity-check [1] ContentIntegrityCheck OPTIONAL,  
    edi-application-security-elements [2] EDIApplicationSecurityElementsField OPTIONAL,  
    security-extensions             [3] SecurityExtensionsField OPTIONAL }  
  
tySecurityExtensionsFieldxe "ty:SecurityExtensionsField"§ ::= SET OF SecurityExtensionsSubField  
  
tySecurityExtensionsSubFieldxe "ty:SecurityExtensionsSubField"§ ::= ExtensionField
```

NOTE - The EDI Application Security Elements Field is defined in . Content and Content Integrity Check are defined in, respectively, 8.2.1.1.1.37 and 8.2.1.1.1.28 of Recommendation X.411. Security services are available only if the MTA supports secure messaging.

specifies how these fields are filled in.

9.1.6 EDIN Initiator

The EDIN Initiator field can take one of the following values:

- a) "internal-UA" means that the UA generated the EDIN either for local reasons or because the generation had been delegated to it by the user.
- a) "internal-MS" means that the MS generated the EDIN either for local reasons or because the generation had been delegated to it by the user.
- c) "external-UA" means that the generation of the EDIN was requested by the user via the abstract operation Originate EDIN (see).

```
tyEDINInitiatorFieldxe "ty:EDINInitiatorField"§ ::= ENUMERATED {  
    internal-ua (0),  
    external-ua (1),  
    internal-ms (2)}
```

Origination of a Positive Notification implies that Responsibility has been accepted, regardless of the value of this field.

The value of this field shall be consistent with the choice (UA, user, PDAU) of the Reason Code field for NNs and FNs.

NOTE - Physical delivery access unit (abPDAUxe "ab:PDAU"§) is defined in .

9.1.7 Notification Extensions

The Notification Extensions allows for future extensions to the EDIN.

tyNotificationExtensionsFieldxe "ty:NotificationExtensionsField"\$::= SET OF NotificationExtensionsSubField

tyNotificationExtensionsSubFieldxe "ty:NotificationExtensionsSubField"\$::= ExtensionField

There are no extensions to the EDIN defined in this Recommendation.

Extensions shall not be critical in EDINs.

9.2 Positive Notifications

A glPositive Notificationxe "gl:Positive Notification"\$ (abPNxe "ab:PN"\$) is sent by the recipient UA, if and only if the originator has requested one, when Responsibility for the EDIM has been accepted by the UA.

The exact procedures which constitute acceptance of Responsibility are a local matter; for example, the UA may construct the PN as soon as it passes the message to the user or it may wait for an external stimulus from the user that the message has been accepted and therefore can send the PN.

Positive Notification fields are defined and described below.

```
tyPositiveNotificationFieldsxe "ty:PositiveNotificationFields"$ ::= SEQUENCE {  
    pn-common-fields          [0] CommonFields,  
    pn-supplementary-information [1] EDISupplementaryInformation OPTIONAL,  
    pn-extensions             [2] PNextensionsField OPTIONAL }
```

9.2.1 PN Supplementary Information

The PN Supplementary Information field may be used to return further information to the EDIN recipient to clarify the Positive Notification.

NOTE - EDI Supplementary Information Field is defined in .

9.2.2 Positive Notification Extensions

The Positive Notification Extensions allows for future extensions to the PN.

tyPNextensionsFieldxe "ty:PNextensionsField"\$::= SET OF PNextensionsSubField

tyPNextensionsSubFieldxe "ty:PNextensionsSubField"\$::= ExtensionField

There are no extensions to the PN defined in this Recommendation.

Extensions shall not be critical in PNs.

9.3 Negative Notifications

A `glNegative Notification` "gl:Negative Notification"§ (abNNxe "ab:NN"§) is sent by a UA, if and only if the originator has requested one, when it determines that it can neither accept Responsibility, nor forward the EDIM and the EDI Notification Request contained in the EDIM to another UA.

Negative Notification fields are defined and described below.

```
tyNegativeNotificationFieldsxe "ty:NegativeNotificationFields"§ ::= SEQUENCE {  
  nn-common-fields           [0] CommonFields,  
  nn-reason-code             [1] NNReasonCodeField,  
  nn-supplementary-information [2] EDISupplementaryInformation OPTIONAL,  
  nn-extensions              [3] NNExtensionsField OPTIONAL }
```

9.3.1 Negative Notification Reason

The Negative Notification Reason indicates why the subject EDIM could not be passed to the user by the UA originating the EDIN. If any Basic Code field has the value "unspecified", additional information may be carried in any combination of a Diagnostic field or the NN Supplementary Info field. Depending on the security policy in force, the security error diagnostic code may or may not be present.

```
tyNNReasonCodeFieldxe "ty:NNReasonCodeField"§ ::= CHOICE {  
  nn-ua-ms-reason-code [0] NNUAMSReasonCodeField,  
  nn-user-reason-code  [1] NNUserReasonCodeField,  
  nn-pdau-reason-code  [2] NNPDAUReasonCodeField }
```

-- *Negative Notification Reason Codes from an EDI-UA or EDI-MS*

```
tyNNUAMSReasonCodeFieldxe "ty:NNUAMSReasonCodeField"§ ::= SEQUENCE {  
  nn-ua-ms-basic-code [0] NNUAMSBasicCodeField,  
  nn-ua-ms-diagnostic [1] NNUAMSDiagnosticField OPTIONAL }
```

- *Negative Notification Basic Reason Codes from an EDI-UA or EDI-MS. These codes are*
- *those specified in Annex B of Recommendation F.435*
- *for the element of service "EDI Notification Request".*

```

tyNNUAMSBasicCodeFieldxe "ty:NNUAMSBasicCodeField"§ ::= INTEGER{
  ecunspecifiedxe "ec:unspecified"§ (0),
  eccannot-deliver-to-userxe "ec:cannot-deliver-to-user"§ (1),
    -- the EDI Interchange can not be passed on to the user
  ecdelivery-timeoutxe "ec:delivery-timeout"§ (2),
    -- the EDI Interchange could not be passed on to the user within
    -- a specified time limit
  ecmmessage-discardedxe "ec:message-discarded"§ (3),
    -- the UA/MS discarded the message before handoff to user
  ecdsubscription-terminatedxe "ec:subscription-terminated"§ (4),
    -- recipient's subscription terminated after delivery but before
    -- handoff to user
  ecforwarding-errorxe "ec:forwarding-error"§ (5),
    -- EDI Forwarding was attempted, but failed.
  ecsecurity-errorxe "ec:security-error"§ (6)
    -- security error

  -- physical delivery errors indicated by "cannot-deliver-to-user"
} (0..ub-reason-code)

```

- *Negative Notification Diagnostic Codes from an EDI-UA or EDI-MS*

```

tyNNUAMSDiagnosticFieldxe "ty:NNUAMSDiagnosticField"§ ::= INTEGER {
  -- This field may be used to further specify the error signalled in nn-ua-ms-basic-code
  -- Additional information may be indicated in nn-supplementary-information

  -- general diagnostic codes
  ecprotocol-violationxe "ec:protocol-violation"§ (1),
    -- used if the UA detects a protocol error
  ecedim-originator-unknownxe "ec:edim-originator-unknown"§ (2),
  ecedim-recipient-unknownxe "ec:edim-recipient-unknown"§ (3),
  ecedim-recipient-ambiguousxe "ec:edim-recipient-ambiguous"§ (4),
    -- used if the EDIM recipients or originator are not valid
  ecaction-request-not-supportedxe "ec:action-request-not-supported"§ (5),
    -- used when the action requested by the recipient is not performed
  ecedim-expiredxe "ec:edim-expired"§ (6),
    -- used when the expiry date of the received EDIM occurred before the subject EDIM
    -- was successfully passed to the user or forwarded by the EDI-UA
  ecedim-obsoleteuxe "ec:edim-obsolete"§ (7),
    -- used when the EDIM Identifier of the received EDIM was contained in the Obsolete EDIM field
    -- of a previously received EDIM.
  ecduplicate-edimxe "ec:duplicate-edim"§ (8),
    -- used when the same EDIM is received more than once from the same originator
  ecunsupported-extensionxe "ec:unsupported-extension"§ (9),
    -- used if the EDIM contains an extension which is not supported by the UA
  ecincomplete-copy-rejectedxe "ec:incomplete-copy-rejected"§ (10),
    -- used if the EDI-UA does not accept EDIMs with the Incomplete Copy Indication true
  ecedim-too-large-for-applicationxe "ec:edim-too-large-for-application"§ (11),
    -- used if the EDIM cannot be delivered to the user due to length constraints
  -- forwarding error diagnostic codes
  ecforwarded-edim-not-deliveredxe "ec:forwarded-edim-not-delivered"§ (12),
    -- used when an Non-Delivery Report is received for forwarded EDIM
  ecforwarded-edim-delivery-time-outxe "ec:forwarded-edim-delivery-time-out"§ (13),
    -- used when no Delivery Report is received within a given period
  ecforwarding-loop-detectedxe "ec:forwarding-loop-detected"§ (14),
    -- used if the UA receives an EDIM wich contains a previously forwarded EDIM
  ecunable-to-accept-responsibilityxe "ec:unable-to-accept-responsibility"§ (15),
    -- used if the EDI-UA cannot accept or forward responsibility

  -- interchange header diagnostic codes
  ecinterchange-sender-unknownxe "ec:interchange-sender-unknown"§ (16),
    -- used when the UA does not recognizes the interchange-sender of the EDI interchange
  ecinterchange-recipient-unknownxe "ec:interchange-recipient-unknown"§ (17),

```


-- used when the UA cannot find a valid interchange recipient in the Recipient Specifier
ecinvalid-heading-fieldxe "ec:invalid-heading-field"§ (18),
ecinvalid-bodypart-typexe "ec:invalid-bodypart-type"§ (19),
ecinvalid-message-typexe "ec:invalid-message-type"§ (20),
ecinvalid-syntax-idxe "ec:invalid-syntax-id"§ (21),

-- security error diagnostic codes
ecmessage-integrity-failurexe "ec:message-integrity-failure"§ (22),
ecforwarded-message-integrity-failurexe "ec:forwarded-message-integrity-failure"§ (23),
ecunsupported-algorithmxe "ec:unsupported-algorithm"§ (24),
ecdecryption-failedxe "ec:decryption-failed"§ (25),
ectoken-errorxe "ec:token-error"§ (26),
ecunable-to-sign-notificationxe "ec:unable-to-sign-notification"§ (27),
ecunable-to-sign-message-receiptxe "ec:unable-to-sign-message-receipt"§ (28),
ecauthentication-failurexe "ec:authentication-failure"§ (29),
ecsecurity-context-failurexe "ec:security-context-failure"§ (30),
xe "ec:message-sequence-failure"§xe "ec:message-security-labelling-failure"xe "ec:repudiation-failure"xe "ec:proof-of-failure"

-- Negative Notification Reason Codes from a user

```
tyNNUserReasonCodeFieldxe "ty:NNUserReasonCodeField"§ ::= SEQUENCE {
  nn-user-basic-code           [0] NNUserBasicCodeField,
  nn-user-diagnostic           [1] NNUserDiagnosticField OPTIONAL }
```

-- Negative Notification Basic Reason Codes from a user

```
tyNNUserBasicCodeFieldxe "ty:NNUserBasicCodeField"§ ::= INTEGER {
  ecunspecifiedxe "ec:unspecified"§ (0),
  ecsyntax-errorxe "ec:syntax-error"§ (1),
  -- used when the user discovers a syntax error within the EDI interchange
  ecinterchange-sender-unknownxe "ec:interchange-sender-unknown"§ (2),
  ecinterchange-recipient-unknownxe "ec:interchange-recipient-unknown"§ (3),
  -- used when the UA cannot find a valid interchange recipient in the Recipient Specifier
  ecinvalid-heading-fieldxe "ec:invalid-heading-field"§ (4),
  ecinvalid-bodypart-typexe "ec:invalid-bodypart-type"§ (5),
  ecinvalid-message-typexe "ec:invalid-message-type"§ (6),
  ecfunctional-group-not-supportedxe "ec:functional-group-not-supported"§ (7),
  ecsubscription-terminatedxe "ec:subscription-terminated"§ (8),
  -- unknown to EDIMS-User service
  ecno-bilateral-agreementxe "ec:no-bilateral-agreement"§ (9),
  ecuser-defined-reasonxe "ec:user-defined-reason"§ (10)
} (0..ub-reason-code)
```

-- Negative Notification Diagnostic Reason Codes from a user

```
tyNNUserDiagnosticCodeFieldxe "ty:NNUserDiagnosticCodeField"§ ::= INTEGER
-- Contains reason passed by user when the value of nn-user-basic-code is user-defined-reason.
-- Additional information may be indicated in nn-supplementary-information
```

-- Negative Notification Reason Codes from a PDAU

```
tyNNPDAUReasonCodeFieldxe "ty:NNPDAUReasonCodeField"§ ::= SEQUENCE {
  nn-pdau-basic-code           [0] NNPDAUBasicCodeField,
  nn-pdau-diagnostic           [1] NNPDAUDiagnosticField OPTIONAL }
```

-- Negative Notification Basic Reason Codes from a PDAU

```
tyNNPDAUBasicCodeFieldxe "ty:NNPDAUBasicCodeField"$ ::= INTEGER {
    ecunspecifiedxe "ec:unspecified"$ (0),
    ecundeliverable-mailxe "ec:undeliverable-mail"$ (1),
        -- used if the PDAU determines that it cannot perform physical delivery of the EDIM
    ecphysical-rendition-not-performedxe "ec:physical-rendition-not-performed"$ (2)
        -- used if the PDAU cannot perform the physical rendition of the EDIM
    } (0..ub-reason-code)
```

-- Negative Notification Diagnostic Codes from a PDAU

```
tyNNPDAUDiagnosticFieldxe "ty:NNPDAUDiagnosticField"$ ::= INTEGER {
    -- This field may be used to further specify the error signalled in nn-pdau-basic-code
    -- Additional information may be indicated in the nn-supplementary-information
    ecundeliverable-mail-physical-delivery-address-incorrectxe "ec:undeliverable-mail-physical-delivery-
address-incorrect"$ (1),
    ecundeliverable-mail-physical-delivery-office-incorrect-or-invalidxe "ec:undeliverable-mail-physical-
delivery-office-incorrect-or-invalid"$ (2),
    ecundeliverable-mail-physical-delivery-address-incompletxe "ec:undeliverable-mail-physical-delivery-
address-incomplete"$ (3),
    ecundeliverable-mail-recipient-unknownxe "ec:undeliverable-mail-recipient-unknown"$ (4),
    ecundeliverable-mail-recipient-deceasedxe "ec:undeliverable-mail-recipient-deceased"$ (5),
    ecundeliverable-mail-recipient-refused-to-acceptxe "ec:undeliverable-mail-recipient-refused-to-accept"$
(6),
    ecundeliverable-mail-organization-expiredxe "ec:undeliverable-mail-organization-expired"$ (7),
    ecundeliverable-mail-recipient-did-not-claimxe "ec:undeliverable-mail-recipient-did-not-claim"$ (8),
    ecundeliverable-mail-recipient-changed-address-permanentlyxe "ec:undeliverable-mail-recipient-changed-
address-permanently"$ (9),
    ecundeliverable-mail-recipient-changed-address-temporarilyxe "ec:undeliverable-mail-recipient-changed-
address-temporarily"$ (10),
    ecundeliverable-mail-recipient-changed-temporary-addressxe "ec:undeliverable-mail-recipient-changed-
temporary-address"$ (11),
    ecundeliverable-mail-new-address-unknownxe "ec:undeliverable-mail-new-address-unknown"$ (12),
    ecundeliverable-mail-recipient-did-not-want-forwardingxe "ec:undeliverable-mail-recipient-did-not-want
forwarding"$ (13),
    ecundeliverable-mail-originator-prohibited-forwardingxe "ec:undeliverable-mail-originator-prohibited-
forwarding"$ (14),
    ecphysical-rendition-attributes-not-supportedxe "ec:physical-rendition-attributes-not-supported"$ (15)
    } (1..ub-reason-code)
```

9.3.2 NN Supplementary Information

The NN Supplementary Information field may be used to return further information to the EDIN recipient to clarify the Negative Notification.

NOTE - EDI Supplementary Information is defined in .

9.3.3 Negative Notification Extensions

The Negative Notification Extensions allows for future extensions to the NN.

```
tyNNExtensionsFieldxe "ty:NNExtensionsField"$ ::= SET OF NNExtensionsSubField
```

tyNNExtensionsSubFieldxe "ty:NNExtensionsSubField"\$::= ExtensionField

There are no extensions to the NN defined in this Recommendation.

Extensions shall not be critical in NNs.

9.4 Forwarded Notifications

A glForwarded Notificationxe "gl:Forwarded Notification"\$ (abFNxe "ab:FN"\$) is sent by a UA, if and only if the originator has requested one, when it determines that it cannot accept Responsibility and decides to forward the EDIM, and the EDI Notification Requests contained in the EDIM, to another UA.

Forwarded Notification fields are defined and described below.

```
tyForwardedNotificationFieldsxe "ty:ForwardedNotificationFields"$ ::= SEQUENCE {  
    fn-common-fields           [0] CommonFields,  
    forwarded-to               [1] ForwardedTo,  
    fn-reason-code             [2] FNReasonCodeField,  
    fn-supplementary-information [3] EDISupplementaryInformation OPTIONAL,  
    fn-extensions              [4] FNextensionsField OPTIONAL }
```

9.4.1 Forwarded To

The Forwarded To field indicates the new recipient of the (forwarded) subject EDIM. Its value is an OR Name.

tyForwardedToxe "ty:ForwardedTo"\$::= ORName

NOTE - OR Name is defined in 8.5.5 of Recommendation X.411.

9.4.2 Forwarded Reason Code

The Forwarded Reason Code indicates the reason why the Responsibility of the subject EDIM was forwarded. If any Basic Code field has the value "unspecified", additional information may be carried in any combination of a Diagnostic field or the FN Supplementary Info field.

```
tyFNReasonCodeFieldxe "ty:FNReasonCodeField"$ ::= CHOICE {  
    fn-ua-ms-reason-code [0] FNUAMSReasonCodeField,  
    fn-user-reason-code  [1] FNUUserReasonCodeField,  
    fn-pdau-reason-code  [2] FNPDAUReasonCodeFields }
```

-- Forwarding Notification Reason Codes from an EDI-UA or EDI-MS

```
tyFNUAMSReasonCodeFieldxe "ty:FNUAMSReasonCodeField"$ ::= SEQUENCE {  
    fn-ua-ms-basic-code [0] FNUAMSBasicCodeField,  
    fn-ua-ms-diagnostic [1] FNUAMSDiagnosticField OPTIONAL,  
    fn-security-check   [2] FNUAMSSecurityCheckField DEFAULT FALSE }
```

-- Forwarding Notification Basic Reason Codes from an EDI-UA or EDI-MS

```
tyFNUAMSBasicCodeFieldxe "ty:FNUAMSBasicCodeField"$ ::= INTEGER {
  ecunspecifiedxe "ec:unspecified"$ (0),
  econward-routingxe "ec:onward-routing"$ (1),
    -- used whenever the UA decides to re-route the subject EDIM for local reasons
  ecrecipient-unknownxe "ec:recipient-unknown"$ (2),
  ecoriginator-unknownxe "ec:originator-unknown"$ (3),
  ecforwarded-by-edi-msxe "ec:forwarded-by-edi-ms"$ (4)
} (0..ub-reason-code)
```

-- Forwarding Notification Diagnostic Reason Codes from an EDI-UA or EDI-MS

```
tyFNUAMSDiagnosticFieldxe "ty:FNUAMSDiagnosticField"$ ::= INTEGER {
  -- This field may be used to further specify the error signalled in fn-ua-ms-basic-code.
  -- Additional information may be indicated in fn-supplementary-information.
  ecrecipient-name-changedxe "ec:recipient-name-changed"$ (0),
  ecrecipient-name-deletedxe "ec:recipient-name-deleted"$ (1)
} (1..ub-reason-code)
```

-- Forwarding Notification Security Check Codes from an EDI-UA or EDI-MS

```
tyFNUAMSSecurityCheckFieldxe "ty:FNUAMSSecurityCheckField"$ ::= BOOLEAN
```

-- Forwarding Notification Reason Codes from a user

```
tyFNUserReasonCodeFieldxe "ty:FNUserReasonCodeField"$ ::= SEQUENCE {
  fn-user-basic-code [0] FNUserBasicCodeField,
  fn-user-diagnostic [1] FNUserDiagnosticField OPTIONAL }
```

-- Forwarding Notification Basic Reason Codes from a user

```
ty FNUserBasicCodeFieldxe "ty: FNUserBasicCodeField"$ ::= INTEGER {
  ecunspecifiedxe "ec:unspecified"$ (0),
  ecforwarded-for-archivingxe "ec:forwarded-for-archiving"$ (1),
  ecforwarded-for-informationxe "ec:forwarded-for-information"$ (2),
  ecforwarded-for-additional-actionxe "ec:forwarded-for-additional-action"$ (3),
  ecsubscription-changedxe "ec:subscription-changed"$ (4),
  echeading-field-not-supportedxe "ec:heading-field-not-supported"$ (5),
  ecbodypart-type-not-supportedxe "ec:bodypart-type-not-supported"$ (6),
  ecmessage-type-not-supportedxe "ec:message-type-not-supported"$ (7),
  ecsyntax-identifier-not-supportedxe "ec:syntax-identifier-not-supported"$ (8),
  ecinterchange-sender-unknownxe "ec:interchange-sender-unknown"$ (9),
  ecinterchange-sender-unknownxe "ec:interchange-sender-unknown"$ (10),
  ecuser-defined-reasonxe "ec:user-defined-reason"$ (11)
} (0..ub-reason-code)
```

-- Forwarding Notification Diagnostic Reason Codes from a user

```
tyFNUserDiagnosticFieldxe "ty:FNUserDiagnosticField"$ ::= INTEGER
  -- Contains reason passed by user when value of fn-user-basic-code is user-defined-reason.
  -- Additional information may be indicated in fn-supplementary-information.
```

-- Forwarding Notification Reason Codes from a PDAU

```
tyFNPDAUReasonCodeFieldxe "ty:FNPDAUReasonCodeField"$ ::= SEQUENCE {
  fn-pdau-basic-code [0] FNPDAUBasicCodeField,
  fn-pdau-diagnostic [1] FNPDAUDiagnosticField OPTIONAL }
```

-- Forwarding Notification Basic Reason Codes from a PDAU

```
tyFNPDAUBasicCodeFieldxe "ty:FNPDAUBasicCodeField"$ ::= INTEGER {  
    ecunspecifiedxe "ec:unspecified"$ (0),  
    ecforwarded-for-physical-rendition-and-deliveryxe "ec:forwarded-for-physical-rendition-and-delivery"$  
    (1)  
} (0..ub-reason-code)
```

-- Forwarding Notification Diagnostic Reason Codes from a PDAU

```
tyFNPDAUDiagnosticFieldxe "ty:FNPDAUDiagnosticField"$ ::= INTEGER
```

A physical delivery access unit (PDAU) (see) is only able to generate NNs and FNs. If PN notification is requested, the PDAU shall generate an FN with appropriate Forwarded Reason Code ("forwarded-for-physical-rendition-and-delivery") when it has determined that it can render the EDIM for physical delivery.

9.4.3 FN Supplementary Information

The FN Supplementary Information field may be used to return further information to the EDIN recipient to clarify the Forwarded Notification.

NOTE - The EDI Supplementary Information Field is defined in .

9.4.4 Forwarded Notification Extensions

The Forwarded Notifications Extensions allows for future extensions to the FN.

```
tyFNExtensionsFieldxe "ty:FNExtensionsField"$ ::= SET OF FNExtensionsSubField
```

```
tyFNExtensionsSubFieldxe "ty:FNExtensionsSubField"$ ::= ExtensionField
```

There are no extensions to the FN defined in this Recommendation.

Extensions shall not be critical in FNs.

10. Primary Object Types

The environment in which EDI Messaging takes place can be modelled as an abstract object which is hereafter referred to as the glEDI Messaging Environmentxe "gl:EDI Messaging Environment"\$ (abEDIMExe "ab:EDIME"\$).

```
vaedimeXe "va:edime"$ OBJECT ::= id-ot-edime
```

When refined (i.e., functionally decomposed), the EDIME can be seen to comprise lesser objects which interact by means of ports.

```

vaedime-refinementxe "va:edime-refinement"$ REFINE edime AS
  edims
    origination [S] PAIRED WITH edimg-user
    reception [S] PAIRED WITH edimg-user
  edimg-user RECURRING
  ::= id-ref-primary

```

The lesser objects are referred to as the glprimary objectxe "gl:primary object"\$s of EDI Messaging. They include a single, central object, the EDI Messaging System, EDIMS, and numerous peripheral objects called EDI Messaging System users (users).

The structure of the EDIME is depicted in Figure .

Error! Cannot open file.§

FIGURE

The EDI Messaging Environmenttc "The EDI Messaging Environment" \f F \l 2§

The primary object types are defined and described below. The types of ports by means of which they interact are discussed in .

10.1 EDI Messaging User

An glEDI Messaging userxe "gl:EDI Messaging user"\$ (abEDIMG userxe "ab:EDIMG user"\$) is typically a computer process or glapplicationxe "gl:application"\$ that engages in EDI Messaging. Such processes or applications are referred to by the term "user" in this Recommendation. A user originates, receives, or both originates and receives Information Objects of the types defined in .

```

vaedimg-userxe "va:edimg-user"$ OBJECT
  PORTS {
    origination [C],
    reception [C]}
  ::= id-ot-edimg-user

```

The EDIME comprises any number of Users.

NOTE - EDI messaging is typically an activity between information processing systems. These are referred to as glEDI applicationsxe "gl:EDI applications"\$. This does not preclude the possibility of human interaction with the information processing systems which are performing EDI, or more direct interaction of a human user with the EDIMS. The terms "gluserxe "gl:user"\$" and "glEDIMG userxe "gl:EDIMG user"\$" may be regarded as synonyms for EDI applications within this Recommendation. For brevity, the term "user" is used throughout the rest of this Recommendation with the meaning of "EDIMG user".

10.2 EDI Messaging System

The glEDI Messaging Systemxe "gl:EDI Messaging System"\$ (abEDIMSxe "ab:EDIMS"\$) is the object by means of which all users communicate with one another in EDI Messaging.

```

vaedimsxe "va:edims"$ OBJECT
  PORTS {
    origination [S],
    reception [S] }
  ::= id-ot-edims

```

The EDIME comprises exactly one EDIMS.

11. Primary Port Types

The primary objects of EDI Messaging are joined to and interact with one another by means of ports. These ports, which the EDIMS supplies, are referred to as the glprimary portxe "gl:primary port"\$s of EDI Messaging. They are of the types defined below.

NOTE - In to follow, the EDIMS is decomposed into still lesser objects, among which is the MTS. This fact is anticipated here by the inclusion of certain MTS capabilities in the EDIMS Abstract Service.

11.1 Origination Port

An glOrigination Portxe "gl:Origination Port"\$ is the means by which a single user conveys to the EDIMS messages containing Information Objects of the types defined in . Through such a port the user originates EDI Messages and EDI Notifications. In addition, the user may originate probes through such a port.

```

vaoriginationxe "va:origination"$ PORT
  CONSUMER INVOKES {
    OriginateProbe,
    OriginateEDIM,
    OriginateEDIN }
  ::= id-pt-origination

```

The EDIMS supplies one Origination Port to each user (with the exception of indirect users served by PDAUs (see).

11.2 Reception Port

A glReception Portxe "gl:Reception Port"\$ is the means by which the EDIMS conveys to a single user messages containing Information Objects of the types defined in . Through such a port the user receives EDI Messages and EDI Notifications. In addition, the user may receive reports through such a port.

```

vareceptionxe "va:reception"$ PORT
  SUPPLIER INVOKES {
    ReceiveReport,
    ReceiveEDIM,
    ReceiveEDIN }
  ::= id-pt-reception

```

The EDIMS supplies one Reception Port to each user.

12. Abstract Operations

What follows defines the abstract service that characterizes EDI messaging, and describes the environment in which that service is supplied and consumed. It does both using the abstract service definition conventions of Recommendation X.407.

The glEDIMS Abstract Servicexe "gl:EDIMS Abstract Service"§ is the set of capabilities that the EDIMS provides to each user by means of one Origination Port and one Reception Port. Those capabilities are modelled as abstract operations, which may encounter abstract errors when invoked.

The purpose of the EDIMS Abstract Service definition is not to prescribe the interface between the EDI user and the EDI-UA, but rather to clarify the meaning and intended use of the Information Objects of . A user interface need not provide commands in one-to-one correspondence to the service's abstract operations, nor indeed even divide the labour between the user and the EDIMS as the service does.

The abstract operations available at the Origination Port and Reception Port are defined and described below. The abstract errors they may provoke are the subject of .

The EDIMS Abstract Service involves neither abstract bind nor abstract unbind operations.

The EDIMS authenticates (i.e., establishes the identity of) the typical user before offering the EDIMS Abstract Service to him. By this means it can verify, e.g., that the user is an EDIMS subscriber. Authentication, where required, is implicit (rather than explicit) in the definition of the EDIMS Abstract Service.

NOTE - In to follow, the EDIMS is decomposed into objects among which is the MTS. The text here reflects this fact by its inclusion of various MTS-defined information items in the EDIMS Abstract Service.

12.1 *Origination Abstract Operations*

The abstract operations available at an origination port are invoked by the user and performed by the EDIMS.

12.1.1 *Originate Probe*

The otOriginate Probexe "ot:Originate Probe"§ abstract operation originates a probe concerning (a class of) messages whose contents are EDIMs.


```

tyOriginateProbex "ty:OriginateProbe"$ ::= ABSTRACT-OPERATION
ARGUMENT SET {
  envelope          [0] ProbeSubmissionEnvelope,
  content            [1] EDIM }
RESULT SET {
  submission-identifier [0] ProbeSubmissionIdentifier,
  submission-time       [1] ProbeSubmissionTime }
ERRORS { RecipientImproperlySpecified }

```

This abstract operation has the following arguments:

- a) otEnvelopex "ot:Envelope"\$: A probe submission envelope, whose make-up the MTS Abstract Service defines. The UA supplies all but the following envelope components, which the user provides:
 - 1) The desired per-message options (i.e., per-message indicators and extensions).
 - 2) The OR Names of the preferred recipients and the per-recipient options (i.e., originator report request, explicit conversion, and extensions) desired for each.
- b) otContentx "ot:Content"\$: An instance of the class of EDIM whose deliverability is to be probed.

This abstract operation has the following results:

- c) otSubmission-identifierx "ot:Submission-identifier"\$: The probe submission identifier the MTS assigns to the probe.
- d) otSubmission-timex "ot:Submission-time"\$: The date and time the probe was directly submitted.

12.1.2 Originate EDIM

The otOriginate EDIMx "ot:Originate EDIM"\$ abstract operation originates a message whose content is an EDIM.

```

tyOriginateEDIMx "ty:OriginateEDIM"$ ::= ABSTRACT-OPERATION
ARGUMENT SET {
  envelope          [0] MessageSubmissionEnvelope,
  content            [1] EDIM }
RESULT SET {
  submission-identifier [0] MessageSubmissionIdentifier,
  submission-time       [1] MessageSubmissionTime }
ERRORS { RecipientImproperlySpecified }

```

This abstract operation has the following arguments:

- a) otEnvelopex "ot:Envelope"\$: A message submission envelope, whose make-up the MTS Abstract Service defines. The UA supplies all but the following envelope components, which the user provides:

- 1) The desired per-message options (i.e., priority, per-message indicators, deferred delivery time, and extensions).
 - 2) The OR Names of the preferred recipients and the per-recipient options (i.e., originator report request, explicit conversion, and extensions) desired for each.
- b) otContentxe "ot:Content"§: The EDIM being originated.
- 1) If application to application security services are required, the user shall supply the value for the EDI Application Security Elements field.

The EDIM shall be constructed as described in .

This abstract operation has the following results:

- c) otSubmission-identifierxe "ot:Submission-identifier"§: The message submission identifier the MTS assigns to the submission.
- d) otSubmission-timexe "ot:Submission-time"§: The date and time the message was directly submitted.

12.1.3 Originate EDIN

The otOriginate EDINxe "ot:Originate EDIN"§ abstract operation originates a message whose content is an EDIN.

```
tyOriginateEDINxe "ty:OriginateEDIN"§ ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope          [0] MessageSubmissionEnvelope,
    content            [1] EDIN }
  RESULT SET {
    submission-identifier [0] MessageSubmissionIdentifier,
    submission-time      [1] MessageSubmissionTime }
  ERRORS { RecipientImproperlySpecified }
```

A user may, if notifications are requested, invoke an Originate EDIN abstract operation to indicate to the UA that it should accept, refuse or forward Responsibility for the subject EDIM. The exact type of EDIN to be generated (PN, NN or FN) is determined from the Content argument.

An EDIN shall be originated only by an actual recipient of the subject EDIM of whom an EDIN is requested by means of the EDI Notification Request field of the subject EDIM's Recipient field.

A user may delegate the task of generating EDINs to the UA. In this case, this abstract operation is not present at the abstract interface between the UA and the user, that is, the operation is not available at the Origination Port. In this case the UA behaves as described in .

This abstract operation has the following arguments:

- a) otEnvelopex "ot:Envelope"§: A message submission envelope, whose make-up the MTS Abstract Service defines. The UA supplies all but the following envelope components, which the user provides:
 - 1) The desired per-message options (i.e., priority, per-message indicators, and extensions). Implicit conversion shall be prohibited, priority shall be that of the subject EDIM.
 - 2) The OR Names of the preferred recipients and the per-recipient options (i.e., explicit conversion and extensions) desired for each. The preferred recipient of the EDIN is the originator of the subject EDIM or, if present, the OR Name indicated in the EDIN Receiver field.
- b) otContentxe "ot:Content"§: The EDIN being originated.
 - 1) If application to application security services are required, the user shall supply the value for the EDI Application Security Elements field.

The EDIN shall be constructed as described in .

This abstract operation has the following results:

- c) otSubmission-identifierxe "ot:Submission-identifier"§: The message submission identifier the MTS assigns to the submission.
- d) otSubmission-timexe "ot:Submission-time"§: The date and time the message was directly submitted.

12.2 Reception Abstract Operations

The abstract operations available at a Reception Port are invoked by the EDIMS and performed by the user.

As abstractly defined, the EDIMS provides no storage for received messages because whether or not it does so for a particular user has no impact upon that user's ability to communicate with other users. Thus the provision of storage is a local matter.

12.2.1 Receive Report

The otReceive Reportxe "ot:Receive Report"§ abstract operation receives a report.

```

tyReceiveReportxe "ty:ReceiveReport"§ ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope                                [0] ReportDeliveryEnvelope,
    undelivered-object                      [1] InformationObject OPTIONAL }
  RESULT
  ERRORS {}

```

The report received may concern any of the following previously originated by the report's recipients:

- a) A message whose content was an EDIM that was originated with the Originate EDIM abstract operation or by forwarding.
- b) A message whose content was an EDIN that was originated as a result of a previously received message. The EDIN could be any of PN, NN or FN.
- c) A probe concerning a message whose content would have been an EDIM that was originated with the Originate Probe abstract operation.

This abstract operation has the following arguments:

- d) `otEnvelopexe "ot:Envelope"§`: A report delivery envelope, whose make-up the MTS Abstract Service defines.
- e) `otUndelivered-objectxe "ot:Undelivered-object"§`: The content of the message whose status is being reported. An EDIM or EDIN.

If the report was provoked by a previous Originate Probe abstract operation invocation, this conditional argument shall be absent. If the report was provoked by a previous Originate EDIM abstract operation invocation, the argument shall be present if, and only if, content return was requested.

This abstract operation has no results.

12.2.2 *Receive EDIM*

The `otReceiveEDIMxe "ot:ReceiveEDIM"§` abstract operation receives a message whose content is an EDIM.

```

tyReceiveEDIMxe "ty:ReceiveEDIM"§ ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope           [0] MessageDeliveryEnvelope,
    content             [1] EDIM }
  RESULT
  ERRORS {}

```

This abstract operation has the following arguments:

- a) `otEnvelopexe "ot:Envelope"§`: The message's delivery envelope.
- b) `otContentxe "ot:Content"§`: The EDIM that is the message's content.

This abstract operation has no results.

When the received EDIM contains an EDIM Body Part (that is, when the original EDIM has been forwarded), it may be necessary to scan several levels of nested Heading fields in order to determine the correct original value for optional Heading fields (see for the nested structure of a forwarded EDIM and for rules related to Heading fields).

12.2.3 *Receive EDIN*

The `otReceiveEDINxe "ot:ReceiveEDIN"$` abstract operation receives a message whose content is an EDIN. The EDIN is provoked by an EDIM originated with the Originate EDIM abstract operation.

```

tyReceiveEDINxe "ty:ReceiveEDIN"$ ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope           [0] MessageDeliveryEnvelope,
    content            [1] EDIN }
  RESULT
  ERRORS {}

```

This abstract operation has the following arguments:

- a) `otEnvelopexe "ot:Envelope"$`: The message's delivery envelope.
- b) `otContentxe "ot:Content"$`: The EDIN that is the message's content.

This abstract operation has no results.

13. Abstract Errors

The abstract errors that may be reported in response to the invocation of the abstract operations available at the Origination Port and Reception Port are defined and described below or as part of the MTS Abstract Service definition.

The set of abstract errors represented below is intended to be illustrative, rather than exhaustive.

13.1 *Recipient Improperly Specified*

The `otRecipient Improperly Specifiedxe "ot:Recipient Improperly Specified"$` abstract error reports that one or more of the OR Names supplied as arguments of the abstract operation whose performance is aborted, or as components of its arguments, are invalid.

This abstract error is defined by the MTS Abstract Service.

14. Other capabilities

In addition to the capabilities embodied in the EDIMS Abstract Service, defined above, the EDIMS shall transparently extend to each user the other MS (see Recommendation X.413) and MTS (see Recommendation X.411) capabilities identified below. (The enumeration of these capabilities necessarily anticipates the fact, stated in , that MSs and the MTS are among the EDIMS' component parts.)

The following additional capabilities shall be provided:

- a) Submission: Capabilities of the MS' or MTS' submission port not embodied in the EDIMS Abstract Service, e.g., the ability to cancel delivery of a previously originated message whose content is an EDIM (but not an EDIN), if deferred delivery was selected.
- b) Delivery: Capabilities of the MTS' delivery port not embodied in the EDIMS Abstract Service, e.g., the ability to temporarily control the kinds of information objects the MTS conveys to the user's UA.
- c) Administration: The capabilities of the MS's or MTS's administration port.
- d) Retrieval: The capabilities of the MS' retrieval port.

In addition to the above and as a local matter, the EDIMS may provide to users additional capabilities neither defined nor limited by this Recommendation. Among such capabilities are those of the Directory.

NOTE - The required capabilities above are excluded from the formal definition of the EDIMS Abstract Service for purely pragmatic reasons, in particular, because their inclusion would largely and needlessly reproduce the definitions of the MS and MTS abstract operations upon which the capabilities are based.

15. Secondary Object Types

The EDIMS can be modelled as comprising lesser objects which interact with one another by means of (additional) ports.

```
vaedims-refinementxe "va:edims-refinement" § REFINE edims AS
  mTS
    submission [S] PAIRED WITH edi-ua, edi-ms
    delivery [S] PAIRED WITH edi-ua, edi-ms
    administration [S] PAIRED WITH edi-ua, edi-ms
  edi-ua RECURRING
    origination [S] VISIBLE
    reception [S] VISIBLE
  edi-ms RECURRING
    submission [S] PAIRED WITH edi-ua
    retrieval [S] PAIRED WITH edi-ua
    administration [S] PAIRED WITH edi-ua
  pdau RECURRING
    reception [S] VISIBLE
  ::= id-ref-secondary
```

These lesser objects are referred to as the glsecondary objectxe "gl:secondary object"\$s of EDI Messaging. They include a single, central object, the MTS, and numerous peripheral objects: EDI messaging system user agents (abEDI-UAxe "ab:EDI-UA"\$), EDI messaging system message stores (abEDI-MSxe "ab:EDI-MS"\$), telematic agents (abTLMAxe "ab:TLMA"\$), and physical delivery access units (abPDAUxe "ab:PDAU"\$). Specification of the protocol for the TLMA may be the subject for future standardization.

The structure of the EDIMS is depicted in Figure . As shown by the figure, EDI-UAs, EDI-MSs, and PDAUs are the instruments by means of which the EDIMS provides the EDIMS Abstract Service to users.

Error! Cannot open file.\$

FIGURE

The EDI Messaging Systemtc "The EDI Messaging System" \f F \l 2\$

The secondary object types are defined and described below. The types of ports by means of which they interact are discussed in .

The refinement above encompasses all possible interconnections of all possible objects. It ignores the possible absence of objects of a particular type (e.g., PDAU), and specific logical configurations of the MS. The latter are identified in Recommendation X.402.

The MTS supplies import and export ports. However, since those ports are not formally defined (in Recommendation X.411), they are not included in the formal refinement above.

Specification of a management port may be the subject for future standardization.

15.1 EDI User Agent

An glEDI user agentxe "gl:EDI user agent"\$ (abEDI-UAxe "ab:EDI-UA"\$) is a UA tailored so as to better assist a single user to engage in EDI Messaging. It helps him originate, receive, or both originate and receive messages containing Information Objects of the types defined in .

```
vaedi-uaxe "va:edi-ua"$ OBJECT
  PORTS {
    origination          [S],
    reception            [S],
    submission           [C],
    delivery             [C],
    retrieval            [C],
    administration       [C] }
::= id-ot-edi-ua
```

The EDIMS comprises any number of EDIMS UAs.

NOTE - As noted above, the term gluser agentxe "gl:user agent"\$ (abUAxe "ab:UA"\$) is used throughout this Recommendation with the meaning of EDI-UA.

15.2 EDI Message Store

An glEDI message storexe "gl:EDI message store"\$ (abEDI-MSxe "ab:EDI-MS"\$) is an MS tailored so as to better assist a single UA engaged in EDI Messaging. It helps it submit, take delivery of, or both submit and take delivery of messages containing Information Objects of the types defined in .

```
vaedi-msxe "va:edi-ms"$ OBJECT
  PORTS {
    submission          [S],
    retrieval            [S],
    administration      [S],
    submission           [C],
    delivery             [C],
    administration      [C] }
::= id-ot-edi-ms
```

The EDIMS comprises any number of EDIMS MSs.

NOTE - As noted above, the term glmessage storexe "gl:message store"\$ (abMSxe "ab:MS"\$) is used throughout this Recommendation with the meaning of EDI-MS.

15.3 Telematic Agent

A gltelematic agentxe "gl:telematic agent"\$ (abTLMAxe "ab:TLMA"\$) is an AU that helps a single indirect user engage in EDI Messaging from a Telematic terminal, along with that terminal and the network that joins the two. A TLMA helps the user originate, receive, or both originate and receive messages containing Information Objects of the types defined in .

Specification of the protocol for this AU may be the subject for future standardization.

15.4 Physical Delivery Access Unit

In the present context, a Physical Delivery Access Unit (abPDAUxe "ab:PDAU"\$) helps any number of indirect users engage in EDI Messaging by means of a Physical Delivery System (abPDSxe "ab:PDS"\$). It helps them receive (but not originate) messages containing Information Objects of the types defined in .

```
vapdauxe "va:pdau"$ OBJECT
  PORTS {
    reception            [S] }
::= id-ot-pdau
```

The EDIMS comprises any number of PDAUs.

A PDAU consumes import and export ports. However, since those ports are not formally defined (in Recommendation X.411), they are not included in the formal definition of PDAU above.

If notifications are requested, the PDAU shall generate one of the following:

- an FN with appropriate reason code if the PDAU determines that it can render and deliver the EDIM
- a NN with appropriate reason code if the PDAU determines that it cannot render or deliver the EDIM.

The use of the PDAU shall be subject to the requirements of the security policy in force.

15.5 Message Transfer System

In the present context, the glMessage Transfer Systemxe "gl:Message Transfer System"§(abMTSxe "ab:MTS"§) conveys Information Objects of the types defined in between UAs, MSs, and AUs.

The EDIMS comprises a single MTS.

The use of TLMA may be restricted by the security policy in force.

16. Secondary Port Types

The secondary objects of EDI Messaging are joined to and interact with one another by means of ports. These ports, which MSs and the MTS supply, are referred to as the glsecondary portxe "gl:secondary port"§s of EDI Messaging. They are of the types identified below.

The capabilities embodied in one Submission, one Retrieval, and one Administration port constitute the MS Abstract Service. They are defined in Recommendation X.413.

The capabilities embodied in one Submission, one Delivery, and one Administration Port constitute the MTS Abstract Service. They are defined in Recommendation X.411.

NOTE - By means of the abstract bind operation which guards its ports, an MS or the MTS typically authenticates another secondary object before offering its abstract service to that object.

Specification of a management port may be the subject for future standardization.

16.1 Submission Port

In the present context, a Submission Port is the means by which a UA (directly or indirectly) or an MS (directly) submits probes concerning, and messages containing Information Objects of the types defined in .

An MS supplies one Submission Port to its UA.

The MTS supplies one Submission Port to each UA configured without an MS and to each MS.

16.2 Delivery Port

In the present context, a Delivery Port is the means by which a UA or MS takes delivery of reports concerning and messages containing Information Objects of the types defined in .

The MTS supplies one Delivery Port to each UA configured without an MS and to each MS.

16.3 Retrieval Port

In the present context, a Retrieval Port is the means by which a UA retrieves reports concerning and messages containing Information Objects of the types defined in .

An MS supplies one Retrieval Port to its UA.

16.4 Administration Port

In the present context, an Administration Port is the means by which a UA changes information about itself or its user on file with its MS, or a UA or MS changes such information on file with the MTS.

An MS supplies one Administration Port to its UA.

The MTS supplies one Administration Port to each UA configured without an MS and to each MS.

16.5 Import Port

In the present context, an Import Port is the means by which the MTS imports reports concerning and messages containing Information Objects of the types defined in .

The MTS supplies one Import Port to each AU.

16.6 Export Port

In the present context, an Export Port is the means by which the MTS exports probes concerning and messages containing Information Objects of the types defined in .

The MTS supplies one Export Port to each AU.

17. User Agent Operation

A UA must employ the MTS in a particular way in order to (correctly) provide the EDIMS Abstract Service to its user. If the user is equipped with an MS, the latter contributes to the provision of the abstract service and, therefore, is subject to the same rules.

The rules that govern the operation of a UA (and MS) are the subject of what follows. The operation of a TLMA is beyond the scope of this Recommendation.

NOTE - The purpose of what follows is not to dictate or constrain the implementation of a real UA unnecessarily, but rather to specify the effect to be achieved.

17.1 Performance of Origination Operations

A UA shall perform the abstract operations it makes available at its Origination Port as prescribed below.

In the performance of these operations, the UA invokes the following abstract operations of the MTS Abstract Service (which, for what follows, are unqualified as to their source):

- a) Probe Submission
- b) Message Submission

In response to the invocation of these abstract operations, a UA reports abstract errors as appropriate. Specification of the precise circumstances under which each abstract error should be reported is beyond the scope of this Recommendation.

17.1.1 Originate Probe

A UA shall perform the Originate Probe abstract operation by invoking Probe Submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of Probe Submission shall be as follows:

- a) Envelope: The components of this argument that constitute per-probe fields shall be as follows; those not explicitly mentioned below shall be as specified by Originate Probe's Envelope argument:
 - 1) Originator-name: The OR Name of the UA's user.
 - 2) Content-type, Content-length, and Original-encoded-information-types: Determined from Originate Probe's Content argument as specified in - .
 - 3) Content-identifier: Specified or omitted as a local matter.

The components of this argument that constitute per-recipient fields shall be as specified by abstract operation Originate Probe's Envelope argument.

The results of Originate Probe shall be as follows:

- b) Submission-identifier: Probe Submission's Probe-submission-identifier result.
- c) Submission-time: Probe Submission's Probe-submission-time result.

The UA shall ignore all properties of Originate Probe's Content argument other than those mentioned above.

How the UA employs Probe Submission's Content-identifier result is a local matter.

17.1.2 *Originate EDIM*

A UA shall perform the Originate EDIM abstract operation by invoking Message Submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of Message Submission shall be as follows:

- a) Envelope: The components of this argument that constitute per-message fields shall be as follows; those not explicitly mentioned below shall be as specified by Originate EDIM Abstract Operation Envelope argument:
 - 1) Originator-name: The OR Name of the UA's user.
 - 2) Content-type and Original-encoded-information-types: Determined from Originate EDIM's Content argument as specified in and , respectively.
 - 3) Content-identifier: Specified or omitted as a local matter.
 - 4) The security arguments on message submission are subject to the security policy in force. When the security policy specifies the support of Content Integrity Security Service, and when Notification Security is requested, the UA shall generate and submit the content-integrity-check Security Argument as defined in 8.2.1.1.1.28 in Recommendation X.411.

The components of this argument that constitute per-recipient fields shall be as specified by abstract operation Originate EDIM's Envelope argument.

To prevent an unknown number of EDINs from being sent to the original originator of a message in case of forwarding, "DL-expansion-prohibited", if available, may be set to TRUE if any of PN, NN or FN are requested.

- b) Content: Determined from Originate EDIM's Content argument (identified as an EDIM) as specified in .

- 1) If "proof/non-repudiation of Responsibility" notification is requested, the UA shall set the Notification Security field accordingly for each recipient as required.
- 2) If "proof/non-repudiation of content received" notification is requested, the UA shall set the Reception Security field accordingly for each recipient as required.
- 3) If "EDI application security" is requested, the end-to-end application security value shall be conveyed in the EDI Application Security Elements field.
- 4) If "proof/non-repudiation of content originated" is requested, the UA shall submit the message using the "message-origin-authentication-check", or the "content-integrity-check" (possibly in the message token), according to the security policy in force.

NOTE - In case of the use of a notarizing function, the non-repudiation of content service is provided implicitly, and is not reflected in any protocol elements.

The results of Originate EDIM shall be as follows:

- c) Submission-identifier: Message Submission's Message-submission-identifier result.
- d) Submission-time: Message Submission's Message-submission-time result.

How the UA employs Message Submission's Content-identifier result is a local matter.

The inclusion of Message Submission's Extensions result among Originate EDIM's results is proper and may be the subject for future standardization.

17.1.3 *Originate EDIN*

A UA shall perform the Originate EDIN abstract operation, if the UA makes it available to its user, by invoking Message Submission with the arguments indicated below, and by returning to its user the results indicated below.

A user may delegate the task of generating EDINs to the UA. In this case, this abstract operation is not present at the abstract interface between the UA and the user, that is, the operation is not available at the Origination port. In this case the UA behaves as if the abstract operation would have been invoked. The UA may accept Responsibility at will, but shall accept Responsibility when the EDIM is made available to the user, or when it forwards an EDIM with content changed (in this context, "content changed" means that the forwarding UA adds or removes body parts from the forwarding EDIM, in accordance with . The term forwarding EDIM is defined in).

A UA shall perform the Originate EDIN abstract operation by invoking Message Submission with the arguments indicated below, and by returning to its user the results indicated below.

The arguments of Message Submission shall be as follows:

- a) Envelope: The components of this argument that constitute per-message fields shall be as follows; those not explicitly mentioned below shall be as specified by the Originate EDIN abstract operation Envelope argument:
 - 1) Originator-name: The OR Name of the UA's user.
 - 2) Content-type and Original-encoded-information-types: Determined from the EDIN as specified in and , respectively.
 - 3) Content-identifier: Specified or omitted as a local matter.
 - 4) Deferred-delivery-time: Omitted.
 - 5) Priority: Same as that of the subject EDIM.

NOTE - Subject EDIM is defined in .

- b) Content: Determined from Originate EDIN's Content argument (identified as a PN, NN or FN) as specified in .
 - 1) If, in the subject EDIM, Reception Security is set to "non-repudiation" and Notification Security is set to "non-repudiation" and the "content-integrity-check" security argument is present in the delivery envelope of the subject EDIM, then the "content-integrity-check" security argument is copied into the Content Integrity Check field of the EDIN. The UA shall submit the EDIN with a non-repudiable security element "content-integrity-check" (possibly in the message token) or a "message-origin-authentication-check" (depending on the security policy in force).
 - 2) If, in the Subject EDIM, Reception Security is set to "non-repudiation" and Notification Security is set to "non-repudiation" and the "content-integrity-check" security argument is not present in the delivery envelope of the subject EDIM, then the Content of the subject message shall be copied into the Original Content field of the EDIN. The UA shall submit the EDIN with a non-repudiable security element "content-integrity-check" (possibly in the message token) or a "message-origin-authentication-check" (depending on the security policy in force).
 - 3) If, in the Subject EDIM, Notification Security is set to "proof" the UA shall submit the EDIN with the security element "content-integrity-check" (possibly in the message token) or the "message-origin-authentication-check", according to the security policy in force.

- 4) If, in the Subject EDIM, EDI Notification Security is set to "non-repudiation" the UA shall submit the EDIN with a non-repudiable security argument "content-integrity-check" (possibly in the message token) or a "message-origin-authentication-check", according to the security policy in force.
- 5) If the MTA does not support secure messaging, the EDIN shall contain an appropriate Reason Code.

The results of Originate EDIN shall be as follows:

- c) Submission-identifier: Message Submission's Message-submission-identifier result.
- d) Submission-time: Message Submission's Message-submission-time result.

How the UA employs Message Submission's Content-identifier result is a local matter.

17.2 Invocation of Reception Operations

A UA shall invoke the abstract operations available at its Reception Port as specified below.

The UA invokes these operations in response to the MTS' invocation of the following abstract operations of the MTS Abstract Service (which, for what follows, are unqualified as to their source):

- a) Report Delivery
- b) Message Delivery

The abstract operations of a Reception Port report no errors.

17.2.1 Receive Report

Whenever the MTS invokes Report Delivery at a UA's Delivery Port, the UA shall invoke the Receive Report abstract operation with the following arguments:

- a) Envelope: Report Delivery's Envelope argument.
- b) Undelivered-object: Determined from Report Delivery's Returned-content argument as specified in .

How the UA employs the Content-identifier component of Report Delivery's Envelope argument is a local matter.

17.2.2 *Receive EDIM*

Whenever the MTS invokes Message Delivery at a UA's Delivery Port, and its Content argument encodes an EDIM as specified in , the UA shall invoke the Receive EDIM abstract operation with the following arguments:

- a) Envelope: Message Delivery's Envelope argument.
- b) Content: Determined from Message Delivery's Content argument as specified in (but no longer marked as an EDIM).

17.2.3 *Receive EDIN*

Whenever the MTS invokes Message Delivery at a UA's Delivery Port, and its Content argument encodes an EDIN as specified in , the UA shall invoke the Receive EDIN abstract operation with the following arguments:

- a) Envelope: Message Delivery's Envelope argument.
- b) Content: Determined from Message Delivery's Content argument as specified in .

17.3 *Internal procedures*

A UA shall perform as specified below the internal procedures that relate to acceptance of Responsibility, refusal of Responsibility and forwarding. Security elements in any type of notification shall follow the rules of .

A user may instruct its UA to accept or refuse Responsibility of incoming messages based on certain criteria.

In addition, a user may instruct its UA to forward incoming messages based on certain criteria.

Because of forwarding, it is possible for a UA to receive the same EDIM more than once. Mechanisms for detecting such duplicate receptions are not required, but may be a matter of local implementation by the UA. If they exist, and notifications are requested, the UA shall originate an NN. If they do not exist, and notifications are requested, the UA shall originate a PN or FN, as appropriate.

The procedures involve the following abstract operations of the MTS Abstract Service (which, for what follows, are unqualified as to their source):

- a) Message Submission
- b) Message Delivery

As implied by the above, in the course of the procedures, the UA has occasion to invoke Message Submission. What it does with the results of this abstract operation is a local matter.

The UA shall consider as a candidate for each procedure individually every message for which all of the following conditions hold:

- c) The MTS has conveyed the message to the UA by invoking Message Delivery at the UA's Delivery Port.
- d) The UA has not conveyed the message to the user by invoking Receive EDIM at the UA's Reception Port.
- e) The message contains an EDIM (rather than an EDIN).

With reference to item d) above, the message might be detained in the UA, e.g., as might be typical, because of the user's unavailability.

17.3.1 *Acceptance of Responsibility*

A UA shall accept Responsibility when a message is successfully passed from the UA to the user. The UA shall follow the procedures below for each candidate message with respect to whose content the following condition holds:

- a) The EDIM requests a PN by means of the EDI Notification Request field of the subject EDIM's Recipients field.

The UA may forward a message that it has accepted Responsibility for. See also on forwarding.

17.3.1.1 *Construction of PN*

The UA shall construct a PN if, and only if, one is requested by means of the EDI Notification Requests field of the subject EDIM's Recipients field and in accordance with

The PN shall have the following common fields:

- a) Subject EDIM: The Subject EDIM's ThisEDIM field or, if present, the Original EDIM Identifier in the EDIN Receiver field.
- b) EDIN Originator: The OR Name of the UA which submits the EDIN.

- c) First Recipient: the OR Name of the UA which the originator of the EDIM specified as recipient, or, if present, the First Recipient field in the EDIN Receiver field. If the EDIN Receiver field is not present, the First Recipient OR Name is the OR Name of the UA creating the EDIN, unless the MTA has performed redirection. In case of redirection, the correct First Recipient OR Name must be obtained from the Intended Recipient Name field of the delivery envelope (see 8.3.1.1.1.4 of Recommendation X.411).
- d) Notification Time: The current date and time.

17.3.1.2 *Submission of PN*

The UA shall submit the PN above by invoking Message Submission with the following arguments:

- a) Envelope: The components of this argument shall be as prescribed for performance of the Originate EDIN abstract operation with the following exceptions:
 - 1) Priority: As specified by Message Delivery's Envelope argument.
 - 2) Per-message-indicators: A local matter, except that conversion-prohibited shall be among the values specified.
 - 3) Per-recipient-fields: A single field whose Recipient-name component shall be the Originator-name component of Message Delivery's Envelope argument, or if the EDIN Receiver field is present, the EDIN Receiver as specified by the originator of the message.

NOTE - If the OR Name in the EDIN Receiver field is not valid, then the UA cannot submit the EDIN. Procedures to be followed in this case are a local matter.

- b) Content: Determined from the PN as specified in .

17.3.2 *Refusal of Responsibility*

A UA shall refuse to accept Responsibility when a message cannot be successfully passed from the UA to the user. A UA may refuse to accept Responsibility when forwarding was unsuccessful (see c) of). The UA shall follow the procedures below for each candidate message under the following conditions:

- a) The EDIM requests an NN of the UA's user by means of the EDI Notification Requests field of the subject EDIM's Recipients field.
- b) The EDIM is not forwarded.

NOTE - See also on forwarding.

17.3.2.1 Construction of NN

The UA shall construct an NN if, and only if, one is requested by means of the EDI Notification Requests field of the subject EDIM's Recipients field and in accordance with

The NN shall have the common fields prescribed for Construction of PN (see).

The NN shall have the following fields:

- a) Negative Reason Code: The reason why Responsibility for the EDIM was refused.
- b) Optionally, supplementary information that adds information to the reason given.

17.3.2.2 Submission of NN

The UA shall submit the NN (if any) above by invoking Message Submission. Its Envelope argument shall be as prescribed for Acceptance of Responsibility (see), its Content argument determined from the NN as specified in .

NOTE - If the OR Name in the EDIN Receiver field is not valid, then the UA cannot submit the EDIN. Procedures to be followed in this case are a local matter.

17.3.2.3 Handling of received EDIM

The received EDIM for which the UA refuses Responsibility shall not be made available to the user.

17.3.3 EDI Forwarding

The procedures defined in this paragraph describe glEDI Forwarding "gl:EDI Forwarding".

NOTE - For brevity, the term "glforwarding" "gl:forwarding" is used in this Recommendation as a synonym for "EDI Forwarding".

A user may instruct its UA to forward received messages based on local criteria. A user may also instruct its UA to automatically forward requests for notifications together with the forwarded message. A message shall not be forwarded when Responsibility for that message has been refused. When Responsibility for a message has been refused, an NN shall be generated, if and only if one has been requested.

In order to forward an EDIM, the UA creates a new EDIM with a new Heading and in the Primary Body Part encapsulates the received EDIM (Heading and Body) and optionally the envelope of the received message using the body part type EDIM Body Part (see).

Figure illustrates forwarding with an example.

Error! Cannot open file.§

FIGURE

Forwardingtc "Forwarding" \f F \l 2§

The term glsubject EDIMxe "gl:subject EDIM"§ refers to the received EDIM that is being forwarded. The term glforwarding EDIMxe "gl:forwarding EDIM"§ refers to the new EDIM that is being created, and that will include all or part of the subject EDIM, in accordance with .

Unless overridden by specific rules below, or by the requirements of the security policy in force, the following general rules apply to the creation of the Heading fields of the forwarding EDIM:

- All mandatory Heading fields and any optional fields whose values are changed with respect to the values present in the subject EDIM shall be present.
- Heading fields whose values are unchanged and whose ASN.1 type is DEFAULT shall be copied from the subject EDIM Heading to the forwarding EDIM Heading if the field is present in the subject EDIM Heading and the value in the field is other than the value specified as DEFAULT in .
- Other Heading fields need not be copied.

EDI Forwarding is done by the MS if the UA has an MS, otherwise by the UA.

EDI Forwarding may take two forms:

- a) Forwarding of message and Responsibility not accepted.
- b) Forwarding of message and Responsibility accepted.

EDI Forwarding may take place even if no notifications have been requested. This is equivalent to form b) above.

The UA shall, subject to the instructions given by the user, forward messages as follows.

17.3.3.1 Forwarding of message and Responsibility not accepted

Forwarding a message without accepting Responsibility of the message implies the following:

- a) The Primary Body Part of the new message is the content of the subject message unchanged. The delivery envelope of the received EDIM shall be included if security notifications are requested.

- b) If the EDI Notification Requests field of the subject EDIM's Recipients field requests notifications and passing of Responsibility is allowed by the originator, the EDI Notification Request is forwarded unchanged with the new message to one, and only one, of the recipients of the new message. The value of the Responsibility Forwarded field shall be set to TRUE.
- c) If the forwarding fails (i.e a Non delivery report on the forwarded message is returned) within a given period of time (either specified by the originator in Expiry Time or as a local decision in the MS or UA, with priority given to the Expiry Time), the UA may send a Negative Notification, NN, back to the originator.
- d) If the EDI Notification Requests field of the subject EDIM's Recipients field requests FN, an FN EDIN shall be sent back to the specified EDIN Receiver, or to the originator of the EDIM, if no EDIN Receiver is specified.

The delivery envelope of the received message shall be included in the new EDIM if the received EDIM's Primary Body Part is not a Forwarded EDIM.

It is possible to forward a message more than once.

A message may be forwarded to multiple recipients.

The originator of a message may prohibit passing of Responsibility by setting the Responsibility Passing Allowed field to the value FALSE. In this case, if the UA cannot accept Responsibility and NN Notification is requested, the UA shall submit an NN EDIN with appropriate reason code. If the UA cannot accept Responsibility and NN Notification is not requested, then no EDIN shall be submitted.

17.3.3.2 Forwarding of message and Responsibility accepted

Forwarding a message and accepting Responsibility of the message implies the following:

- a) The Primary Body Part of the new message is the content of the subject message changed or unchanged. This type of forwarding is less restricted and may include removal or addition of body parts.

NOTE 1 - If the delivery envelope of the received message is included in the forwarded message, and if that envelope contained security fields, and if body parts are added or removed, then the security fields may no longer be valid.

The following rules apply with respect to removal of body parts:

- 1) A forwarded EDIM Body Part shall not be removed.
- 2) "removed-edi-body" shall be inserted where an EDI Body Part has been removed (see).

- 3) Body Part Place Holders shall be inserted where other body parts have been removed (see).
- 4) the Incomplete Copy Indicator field shall be set to "TRUE" if Body Parts are removed.
- b) If the EDI Notification Requests field of the subject EDIM's Recipients field requests Positive Notification, a PN EDIN shall be sent back to the specified EDIN Receiver, or to the originator of the EDIM, if no EDIN Receiver is specified.
- c) A Forwarded Notification, FN, shall not be sent back to the originator of the message.
- d) The Responsibility Forwarded field shall not be present.

NOTE 2 - By scanning the successive nested Headings of an EDIM that contains a forwarded EDIM, the final recipient UA can determine from the setting of the Responsibility Forwarded field at which point in the forwarding chain Responsibility was accepted.

17.3.3.3 *Prevention of loops*

The UA shall suppress forwarding if, and only if, the EDIM to be forwarded itself contains a forwarding EDIM that the UA previously created. Forwarding shall be suppressed whenever the forwarding EDIM appears (directly) in a body part of the EDIM to be forwarded, or (nested) in a body part of the EDIM that appears in such a body part.

The UA shall consider itself to have created the forwarding EDIM above if, and only if, the OR Name component of a This EDIM Field in a forwarded EDIM matches the OR Name of the UA's user.

NOTE - Forwarding an EDIM of the kind described above would constitute an EDI Forwarding "loop".

17.3.3.4 *Construction of EDIM*

The UA shall construct a forwarding EDIM whose Primary Body Part comprises a body part of type EDIM Body Part. Additionally, other body parts may be added, and body parts may be removed.

The Heading shall have the following components:

- a) This EDIM: New value generated.
- b) Originator: OR Name of the forwarding user.
- c) Recipients: The recipients to which the EDIM shall be forwarded.

- d) Incomplete Copy: If present in the Heading of the subject EDIM.

If Responsibility is not accepted, the following rules relating to the components of the EDIM Heading apply:

- e) EDIN Receiver Field: shall be present and shall contain all optional fields. If the subject EDIM contains an EDIN Receiver Field, the fields of the EDIN Receiver Field of the forwarding EDIM shall have the values of the fields of the EDIN Receiver Field of the subject EDIM. If optional fields are missing from the EDIN Receiver Field of the subject EDIM, or if the subject EDIM does not contain an EDIN Receiver Field, then the fields of the EDIN Receiver Field of the forwarding EDIM shall have the following values:
 - 1) Edin-receiver: Originator of subject EDIM.
 - 2) Original-edim-identifier: This EDIM field of subject EDIM.
 - 3) First-recipient: OR Name of the UA to which the subject EDIM was first sent by the original originator. This is the OR Name of the forwarding UA (which is performing the first forwarding operation), unless the MTA has performed redirection. In case of redirection, the correct First Recipient OR Name must be obtained from the Intended Recipient Name field of the delivery envelope (see 8.3.1.1.1.4 of Recommendation X.411).
- f) EDI Notification Request (sub-field of Recipients): The UA may forward the EDIM to several recipients by simply adding recipients to the Recipients field. If EDI Notification Requests are present in the subject EDIM, the UA shall set identical EDI Notification Requests for one, and only one, of the recipients. One, and only one, of the UAs to whom the subject EDIM is forwarded will receive the EDI Notification Requests contained in the subject EDIM.
- g) Expiry Time: may be set to a value different to the value indicated in the subject EDIM.
- h) All other Heading fields shall follow the general rules in .

If Responsibility is accepted, the EDIM Heading shall comply with a), b) and c) above and with the following rules:

- i) EDIN Receiver Field: may be absent or present. If present, it shall contain only the following value:
 - 1) Edin-receiver: OR Name of the desired EDIN Receiver.
- j) Other fields may be added (including EDI Notification Requests).

Other fields apart from those especially mentioned above may, but need not, be copied from the Heading of the subject EDIM to the Heading of the new EDIM (except that the Original EDIM Identifier and First Recipient sub-fields of the EDIN Receiver Field shall not be present).

NOTE - If several recipients have the same OR Name, and notifications were requested from each, then the original originator may receive EDIN's which have identical Subject EDIM, EDIN Originator, and First Recipient fields.

The Primary Body Part shall be of type EDIM Body Part and shall have the following components:

- k) Parameters: The Envelope argument of Message Delivery.
- l) Data: The EDIM to be forwarded.

17.3.3.5 *Submission of EDIM*

The UA shall submit the EDIM it constructed above by invoking Message Submission with the following arguments:

- a) Envelope: The components of this argument shall be as follows:
 - 1) Originator-name: The OR Name of the UA's user.
 - 2) Content-type and Original-encoded-information-types: Determined from the EDIM as specified in and .
 - 3) Content-identifier: Specified or omitted as a local matter.
 - 4) Priority: As specified by Message Delivery's Envelope argument.
 - 5) Per-message-indicators and Extensions: A local matter.
 - 6) Deferred-delivery-time: Omitted.
 - 7) Per-recipient-fields: Their Recipient-name components shall be the OR Names that the message shall be forwarded to. Their other components are a local matter.
- b) Content: Determined from the EDIM as specified in .

17.3.3.6 *Construction of FN*

The UA shall construct an FN if, and only if, one is requested by means of the EDI Notification Requests field of the subject EDIM's Recipients field and the user is not willing to accept Responsibility for the message and forwards the request for notification.

The FN shall have the common fields as prescribed for acceptance of Responsibility (See).

The FN shall have the following forwarding fields:

- a) Forwarded To: the OR Name of the recipient to whom the request for notification was forwarded.
- b) Forwarded Reason Code: The reason why the subject message was forwarded.
- c) Optionally, supplementary information that adds information to the reason given.

17.3.3.7 *Submission of FN*

The UA shall submit the FN (if any) above by invoking Message Submission. Message Submission's Envelope argument shall be as prescribed for acceptance of Responsibility (See), its Content argument determined from the FN as specified in .

NOTE - If the OR Name in the EDIN Receiver field is not valid, then the UA cannot submit the EDIN. Procedures to be followed in this case are a local matter.

18. Message Store operation

Recommendation X.413 defines the abstract service for a general content independent Message Store (MS). The MS is an optional system component in an MHS. The MS is associated with a user's UA. The user can submit messages through it and retrieve messages that have been delivered to the MS. In addition, the MS can perform certain predefined. auto actions on the UA's behalf.

NOTE - Because the MS is an optional system component in an MHS, use of the word "shall" with respect to MS specifications shall not be construed as mandating the provision of an MS or the services it provides. Use of the word "shall" with respect to MS specifications shall be construed as mandating the specifications of an MS, if one is provided.

All the abstract operations, general attribute types and general auto actions types defined in Recommendation X.413 are also available for use by EDI messages.

An MS may optionally offer additional support for the EDI messaging specific attribute types and auto actions, which would qualify it as an EDI messaging specific MS (EDI MS). These additional definitions are given in what follows.

18.1 *Binding to the MS*

Binding to the MS is described in 7.1 of Recommendation X.413. Attention should be given to the following points when using the MS for EDI messaging.

18.1.1 *Abstract-bind argument*

The following parameters from 7.1.1 of Recommendation X.413 have special meaning in this Recommendation:

- a) Fetch-restrictions
The name of the object identifier for the EDI content type is "id-mct-pedi", the value is defined in Annex A.
- b) Allowed-EITs
The names of the object identifiers so far standardized in this Recommendation are defined in Annex A. See also .

18.2 *Abstract-bind result*

The following parameter from 7.1.2 of Recommendation X.413 has special meaning for this Recommendation:

- Available-auto-actions

NOTE - The use of the general auto action "auto-forward" is discouraged for use with EDIMs. Instead the EDI messaging specific auto actions "edi-forward-with-responsibility-accepted" and "edi-forward-with-responsibility-not-accepted" should be used.

18.3 *Creation of Information Objects*

An MS shall satisfy the following requirements related to the information objects it maintains:

- a) The MS shall maintain a separate information object for each message containing an EDIM or EDIN that is delivered to it.
- b) The MS shall maintain as a separate information object not only each message containing a forwarding EDIM (pursuant to Item a) but also each message containing a forwarded EDIM (recursively).
- c) The MS shall assign sequence numbers to the messages in the hierarchy formed by a forwarding EDIM and its forwarded EDIMs. The lowest number shall be assigned to the outermost level of the hierarchy.

The general (content independent) attributes that may occur in a stored-messages information-base are documented in Recommendation X.413. All content-independent MS attributes can be used for the content defined in this Recommendation. The EDI specific attributes for stored-messages are defined in . All general attribute types classified as "mandatory" in Table 1 of Recommendation X.413 shall be supported.

18.3.1 Mapping of an MHS message in MS

NOTE - In what follows, reference is made to an "MHS message". This is not be confused with the term "message", which refers to an EDIM.

When an EDIM or EDIN gets delivered into the MS, a corresponding MS entry is generated in the stored-messages information-base. The MS generates some attributes for administration purposes such as Sequence Number, a Creation Time for the entry, the Interchange Length etc. It then generates attributes based on protocol elements in the MHS Envelope, in the Heading and one attribute containing the whole EDI Interchange. The attribute EDI Body Part Type signals which EDI Standard has been used. Similarly, other Body Parts will be mapped into one or several additional attributes.

Figure describes how an MHS message with an EDIM is mapped into a corresponding MS entry.

Error! Cannot open file.§

FIGURE

MHS Message with EDIM - Mapping in MStc "MHS Message with EDIM - Mapping in MS" \f F \l 2§

18.3.2 Mapping of forwarded messages in MS

A forwarded EDIM is mapped into the Message Store as one main entry and one or more linked child entries. The final child entry will contain the original EDIM (with its interchange and any additional body parts).

The MS Structure of a forwarded message such as the message in Figure is depicted in Figure .

Error! Cannot open file.§

FIGURE

Forwarded message in MStc "Forwarded message in MS" \f F \l 2§

18.4 Maintenance of Attributes

An MS shall satisfy the following requirements related to MS attributes:

- a) For each EDIM or EDIN it holds, the MS shall support the attributes of as specified therein.
- b) For each EDIM it holds, the MS shall give the following meanings to the defined values of the MS-status attribute:
 - 1) new: No attribute values have been conveyed to the UA.
 - 2) listed: At least one attribute value has been conveyed to the UA, and at least one body part value has not been conveyed to the UA.

- 3) processed: All body parts have been conveyed to the UA or the MS has performed an auto action.
- c) For each EDIN it holds, the MS shall give the following meanings to the defined values of the MS-status attribute:
 - 1) new: No attribute values have been conveyed to the UA.
 - 2) listed: At least one attribute value has been conveyed to the UA, and at least one attribute value has not been conveyed to the UA.
 - 3) processed: All attributes have been conveyed to the UA or the MS has performed an auto action.
- d) The MS-status attribute shall reflect the state of affairs prior to an abstract operation invocation that alters its value.
- e) The Content Type attribute of each message containing an EDIM or EDIN that is delivered to the MS shall have as value the Object Identifier id-mct-edi (see Annex A).

18.5 Negative Notification

When it discards an EDIM while performing the Delete abstract operation of the MS Abstract Service, the MS shall submit an NN if one is requested and the EDIM's MS-status attribute has the value listed.

18.6 Auto Action Types

The concept of auto actions is described in 6.5 and 12 of Recommendation X.413. This defines two general auto action types, which can potentially be used for all content-types. However, the "auto-forward" auto action defined in there is not well suited for the EDIM content-type and its use for EDI messaging is discouraged. Instead, a specific auto action, for EDI Forwarding is defined below.

The auto-alert auto action defined in 12.2 of Recommendation X.413 can be used for EDI messaging without any restrictions.

Auto actions are registered and deregistered using the Register-MS abstract operation as described in 8.6 of Recommendation X.413.

The EDI-auto-forward auto action is described in the following. The operation of this auto action may be affected by the implementation of a security policy.

The auto action is described below together with its abstract syntax using the AUTO-ACTION macro is defined in 6.5 of Recommendation X.413. The definition of EDI-auto-forward auto actions are defined below.

The EDI-auto-forward allows EDIMs to be forwarded as follows:

- forwarding-with-responsibility-not-accepted, which means that the EDI responsibility is forwarded. See a) of ;
- forwarding-with-responsibility-accepted, which means that the EDI responsibility is accepted. See b) of ;

As specified in , if no notifications are requested EDI-auto-forwarding may take place, and is equivalent to forwarding-with-responsibility-accepted.

If EDI Security Requests are present, then the EDI-auto-forward actions defined above may be prohibited, subject to the security policy in force. If EDI Security Requests are present then the EDI-auto-forward action "forwarding-with-responsibility-accepted" shall not be performed.

The EDI-auto-forward allows one or more sets of EDI-forward-registration-parameters to be registered with the MS, each identified by its registration-identifier. Each EDI-forward-registration-parameter specifies criteria to determine whether it applies to a delivered EDIM, and if so, a copy of the message is EDI-auto-forwarded using the Message-submission abstract operation. The delivered EDIM may be automatically deleted afterwards. Below is the ASN.1 definition of the edi-auto-forward AUTO ACTION:

```
moedi-auto-forwardxe "mo:edi-auto-forward"$ AUTO-ACTION
  REGISTRATION PARAMETER IS EDIForwardRegistrationParameter
  ::= id-act-edi-auto-forward

moEDIForwardRegistrationParameterxe "mo:EDIForwardRegistrationParameter"$ ::= SEQUENCE {
  filter                                [0] Filter OPTIONAL,
  edi-supplementary-info                [1] EDISupplementaryInfo OPTIONAL,
  delete-after-forwarding               [2] BOOLEAN DEFAULT FALSE,
  edi-forwarding-mode                   CHOICE {
    forwarding-with-responsibility-not-accepted [3] ForwardWithRespForwarded,
    forwarding-with-responsibility-accepted    [4] ForwardWithRespAccepted }
}
```

NOTE - The data types Filter, Per Message Auto Forward Fields and Per Recipient Auto Forward Fields are defined in 12.1 of Recommendation X.413.

The common parameters of the EDI Forward Registration Parameter have the following meanings:

- a) Filter: This is a set of criteria which a new entry representing a delivered EDIM shall satisfy for the MS abstract service provider to auto forward it using this set of parameters.

The absence of this parameter indicates that all new entries are to be examined for potential auto forwarding.

NOTE - Substrings in filters cannot be defined for composite attributes (attributes with further ASN.1 structure in the attribute value) in the present version of Recommendation X.413.

- b) Edi-supplementary-info: This parameter can contain text to be included in the EDIN supplementary field of an EDIN and in the other-parameters field of a forwarded EDIM.
- c) Delete-after-forwarding: This parameter indicates whether an MS entry should be deleted or not, once the auto-forward submission has succeeded. If not specified, no deletion takes place.
- d) Edi-forwarding-type: This is a choice between:
 - 1) forwarding-with-responsibility-not-accepted,
 - 2) forwarding-with-responsibility-accepted,

The remaining parameters are described separately for the three cases below.

18.6.1 *Forwarding With Responsibility Not Accepted*

The forwarding-with-responsibility-not-accepted case enables the MS abstract service provider to automatically forward, with EDI Responsibility forwarded, any EDIM that has been delivered into the stored-messages information base. The use of this auto action shall be subject to the requirements of the security policy in force. The MS shall follow the rules in . Appropriate values are added in the EDI Notification Indication attribute.

The following limitations apply to forwarding-with-responsibility-not-accepted, when compared to the general rules for forwarding contained in :

- a) The forwarding-with-responsibility-not-accepted auto action type shall only be performed once for a particular EDIM by the same MS.
- b) Only one recipient shall be specified for the forwarding auto action.

```

moForwardWithRespNotAcceptedxe "mo:ForwardWithRespNotAccepted"$ ::= SET {
  COMPONENTS OF PerMessageAutoForwardFields, -- from envelope PerMessageFields
  per-recipient-field                          [3] PerRecipientAutoForwardFields,
  notification-argument                       [4] NotificationArguments OPTIONAL }

moNotificationArgumentsxe "mo:NotificationArguments"$ ::= SET {
  COMPONENTS OF PerMessageAutoForwardFields, -- from envelope PerMessageFields
  per-recipients-field                        [3] SEQUENCE SIZE (1..ub-recipients) OF
                                              PerRecipientAutoForwardFields }

```

The following ASN.1 data type defines the parameters specific to this case:

- c) PerMessageAutoForwardFields:
This is a set of arguments registered to be used for each message submission abstract operation (see 8.2.1.1.1 of Recommendation X.411). Any argument which is not registered, not mandatory, and not specifically mentioned below, will be absent from each message submission.

If the following arguments are either not registered, or registered with their default values, the values used for each message submission abstract operation are those of the corresponding message delivery arguments: priority, implicit-conversion-prohibited, and conversion-with-loss-prohibited.

If the following argument is either not registered, or registered with their default values, their presence as message submission arguments depends upon the presence of the corresponding message delivery arguments, their values being transformed where appropriate: message-security-label.

The following arguments have fixed values:

- 1) DL-expansion-prohibited, value DL-expansion-prohibited
- 2) implicit-conversion-prohibited: value implicit-conversion-prohibited
- 3) conversion-with-loss-prohibited: value conversion-with-loss-prohibited

Certain message submission arguments may be registered. These are: proof-of-submission-request, original-encoded-information-types and content-type.

d) PerRecipientAutoForwardFields:

This is a set of arguments registered to be used for each message submission abstract operation (see 8.2.1.1.1 of Recommendation X.411). Any argument which is not registered, not mandatory, and not specifically mentioned below, will be absent from each message submission.

The following argument have a fixed value:

- 1) originator-report-request: this shall have either the value non-delivery-report or the value report.

Only one recipient is allowed for this case.

e) Notification-argument:

This contains the same parameters as in c) and d) above, but the actual values can differ from the values in the forwarded EDIM.

18.6.2 *Forwarding with responsibility accepted*

The forwarding-with-responsibility-accepted case enables the MS abstract service provider to automatically forward, with Responsibility accepted, any EDIM that has been delivered into the stored-messages information base. The use of this auto action shall be subject to the requirements of the security policy in force. The MS shall follow the rules in . Appropriate values are added in the EDI Notification Indication attribute.

The following limitations apply to forwarding-with-responsibility-accepted, when compared to the general rules for forwarding contained in :

a) The MS shall construct and forward an EDIM whose primary body part comprises a body part of type EDIM body part as described in , however no body parts shall be removed or added, the original delivery envelope shall be included and the components of the original Heading shall be copied to the Heading of the forwarding EDIM according to the rules in ; with the following exceptions:

- 1) The "recipient" parameter value is set to the next "recipient".
- 2) Any registered values for Heading fields shall replace the old values in the new Heading.

```

moForwardWithRespAcceptedxe "mo:ForwardWithRespAccepted"$ ::= SET {
  COMPONENTS OF PerMessageAutoForwardFields, -- from envelope PerMessageFields
  next-recipients-field          [3] SEQUENCE SIZE (1..ub-recipients) OF
                                  PerRecipientAutoForwardFields,
  notification-argument         [4] NotificationArguments OPTIONAL,
  heading-fields                 [5] HeadingFields OPTIONAL }

moHeadingFieldsxe "mo:HeadingFields"$ ::= SEQUENCE {
  new-edin-receiver-name        [0] ORName OPTIONAL,
  next-recipient                 [1] RecipientField,
  next-recipient-action-request  [2] ActionRequestField DEFAULT {id-for-action},
  next-recipient-edi-notification-requests-field [3] EDINotificationRequestsField OPTIONAL,
  next-responsibility-passing-allowed [4] ResponsibilityPassingAllowedField DEFAULT FALSE }

```

The following ASN.1 data type defines the parameters specific to this case:

b) PerMessageAutoForwardFields:
The description is the same as in c) of .

c) PerRecipientAutoForwardFields
The description is the same as in d) of .

Multiple recipients are allowed for this case.

d) Notification-argument:
The description is the same as in e) of .

e) HeadingFields:
New values may be defined for:

- 1) "new-edin-receiver-name" to replace "edin-receiver-name" in EDIN Receiver Field.
- 2) "next-recipient" to replace "recipient" in Recipients Sub Field. This field is mandatory.
- 3) "next-recipient-action-request" to replace "recipient-action-request" in Recipients Sub Field.
- 4) "next-recipient-edi-notification-requests-field" to replace "recipient-edi-notification-requests-field" in Recipients Sub Field.

- 5) "next-responsibility-passing-allowed" to "replace responsibility-passing-allowed" in Recipients Sub Field.

18.7 Message Store Attributes

As described in Recommendation X.413, an MS maintains and provides access to certain attributes of each information object it holds. An attribute comprises a type and, depending upon the type, one or more values. Attributes that may assume several values simultaneously (all pertaining to one object) are termed multi-valued, those that may assume just one value, single-valued. Some attributes pertain to information objects of all kinds, others only to those of e.g. EDI messaging kind.

The following defines the MS attributes specific to EDI messaging. EDI specific attributes are defined below.

All of the attributes defined in this Recommendation, except those corresponding to extended body part types (which cannot be enumerated), are listed alphabetically, for reference, in the first column of Table 1. This table records their presence in a delivered message entry. None of them appears in a delivered report entry. Additional, unnamed attributes are described in . Table 2 describes how the EDI attributes are generated.

NOTE - See and for an elaboration of the legend of the tables.

TABLE 1

Summary of EDI specific MS Attribute Types "Summary of EDI specific MS Attribute Types" ¶ T ¶ 2§

Attribute	Single/ Multi valued UA	Support level by MS and EDIM	Presence in in delivered PN NN	Presence in in delivered FN	Presence for list, delivered	Presence for delivered	Available alerts	Available summarize
acknowledgement-request- for-this-recipient	S O	P -	- -	Y N				
action-request- for-this-recipient	S O	P -	- -	Y N				
application-reference	S O	C -	- -	Y N				
authorization-information- for-this-recipient	S O	C -	- -	Y N				
body	S M	P -	- -	N N				
communications-agreement-id- for-this-recipient	S O	C -	- -	Y N				
cross-referencing-information	M O	C -	- -	Y N				
date-and-time-of-preparation	S M	C -	- -	Y N				
edi-application-security-elements	S O	C -	- -	Y N				
edi-application-security-extensions	M O	C -	- -	Y N				
edi-body-part	S M	P -	- -	N N				
edi-bodypart-type	S M	P -	- -	Y Y				
edi-message-type	M M	C -	- -	Y N				
edi-notification-indicator	M O	- -	- -	Y N				
edi-notification-requests- for-this-recipient	S O	C -	- -	Y N				
edi-notification-security- for-this-recipient	S O	C -	- -	Y N				
edi-reception-security- for-this-recipient	S O	C -	- -	Y N				
edim-body-part	S O	C -	- -	N N				
edim-synopsis	S O	P -	- -	N N				
edims-entry-type	S M	P P	P P	Y Y				
edin-initiator	S O	- P	P P	Y N				
edin-originator	S O	- P	P P	Y N				
edin-receiver	S O	C -	- -	Y N				
expiry-time	S O	C -	- -	Y N				
externally-defined-body-part-types	M O	C -	- -	Y N				
first-recipient	S O	C C	C C	Y N				
fn-extensions	M O	- -	- C	Y N				
fn-reason-code	S O	- -	- P	Y N				
fn-supplementary-information	S O	- -	- C	Y N				
forwarded-to	S O	- -	- P	Y N				
heading	S M	P -	- -	N N				
heading-extensions	M O	C -	- -	Y N				

TABLE 1 (concluded)

Attribute	Single/ Multi valued	UA	Support level by MS and EDIM	Presence in in delivered PN NN	Presence in in delivered FN	Presence for list, delivered	Presence for delivered	Available alerts	Available summarize
incomplete-copy	S	O	P	-	-	-	Y	N	
interchange-control-reference- for-this-recipient	S	M	C	-	-	-	Y	N	
interchange-length	S	O	P	-	-	-	Y	N	
interchange-recipient- for-this-recipient	S	M	C	-	-	-	Y	N	
interchange-sender	S	M	C	-	-	-	Y	N	
message-data	S	O	C	-	-	-	N	N	
message-parameters	S	O	C	-	-	-	N	N	
nn-extensions	M	O	-	-	C	-	Y	N	
nn-reason-code	S	O	-	-	P	-	Y	N	
nn-supplementary-information	S	O	-	-	C	-	Y	N	
notification-security-elements	S	O	-	C	C	C	Y	N	
notification-time	S	O	-	P	P	P	Y	N	
notifications-extensions	M	O	-	C	C	C	Y	N	
obsoleted-edims	M	O	C	-	-	-	Y	N	
originator	S	O	C	-	-	-	Y	N	
pn-extensions	M	O	-	C	-	-	Y	N	
pn-supplementary-information	S	O	-	C	-	-	Y	N	
processing-priority-code- for-this-recipient	S	O	C	-	-	-	Y	Y	
recipient-extensions- for-this-recipient	M	O	C	-	-	-	Y	N	
recipient-reference- for-this-recipient	S	O	C	-	-	-	Y	N	
related-messages	M	O	C	-	-	-	Y	N	
responsibility-forwarded	S	O	P	-	-	-	Y	Y	
responsibility-passing-allowed- for-this-recipient	S	O	P	-	-	-	Y	N	
service-string-advice	S	O	C	-	-	-	Y	N	
subject-edim	S	M	-	P	P	P	Y	N	
syntax-identifier	S	M	C	-	-	-	Y	Y	
test-indicator- for-this-recipient	S	O	P	-	-	-	Y	Y	
this-edim	S	M	P	-	-	-	Y	N	
this-recipient	S	O	P	-	-	-	Y	N	

TABLE 2

Generation of the EDI specific MS Attribute Types "Generation of the EDI specific MS Attribute Types" f T \ 2§

Attribute-type-name	Source parameters by	Source generated	Generation rules
acknowledgement-request-for-this-recipient	acknowledgement-request	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient. If the source parameter is missing, an attribute with the default value shall be generated.
action-request-for-this-recipient	action-request	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient. If the source parameter is missing, an attribute with the default value shall be generated.
application-reference	application-reference	MD	The value of the parameter is the attribute value.
authorization-information-for-this-recipient	authorization-information	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
body	body	MD	The value of the parameter is the attribute value.
communications-agreement-id-for-this-recipient	communications-agreement-id	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
cross-referencing-information	cross-referencing-information	MD	A value is generated from each value of the SET.
date-and-time-of-preparation	date-and-time-of-preparation	MD	The value of the parameter is the attribute value.
edi-application-security-elements	edi-application-security-elements	MD	The value of the parameter is the attribute value.
edi-application-security-extensions	edi-application-security-extensions	MD	A value is generated from each value of the SET.
edi-body-part	edi-body-part	MD	The value of the parameter is the attribute value.
edi-bodypart-type	edi-bodypart-type	MD	The value of the parameter is the attribute value. If the source parameter is missing, an attribute with the default value shall be generated.
edi-message-type	edi-message-type	MD	A value is generated from each value of the SET.
edi-notification-indicator	NONE	MSA	A value is added when an EDIN is submitted from the MS.
edi-notification-requests-for-this-recipient	edi-notification-requests	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
edi-notification-security-for-this-recipient	edi-notification-security	MD	The value of the parameter is the attribute value.

TABLE 2 (continued)

Attribute-type-name	Source parameters by	Source generated	Generation rules
edi-reception-security-for-this-recipient	edi-reception-security	MD	The value of the parameter is the attribute value.
edim-body-part	NONE	MS	The value is the sequence-number of the entry created for the forwarded EDIM.
edim-synopsis	se 18.7.1.2	MS	se 18.7.1.2
edims-entry-type	InformationObject and edin	MS	If the information object is an EDIM, the value is set to "edim". If the information object is an EDIN, the value is set according to the type of the EDIN.
edin-initiator	edin-initiator	MD	The value of the parameter is the attribute value.
edin-originator	edin-originator	MD	The value of the parameter is the attribute value.
edin-receiver	edin-receiver	MD	The value of the parameter is the attribute value.
expiry-time	expiry-time	MD	The value of the parameter is the attribute value.
externally-defined-body-part-types	additional-body-parts	MD	From each component of the SEQUENCE, one value is generated from the value of the ExternallyDefinedData components direct-reference and one is generated from the value of the ExternallyDefinedParameters components direct-reference, if present.
first-recipient	first-recipient	MD	The value of the parameter is the attribute value.
fn-extensions	fn-extensions	MDA	A value is generated from each value of the SET.
fn-reason-code	fn-reason-code	MD	The value of the parameter is the attribute value.
fn-supplementary-information	fn-supplementary-information	MD	The value of the parameter is the attribute value.
forwarded-to	forwarded-to	MD	The value of the parameter is the attribute value.
heading	heading	MD	The value of the parameter is the attribute value.
heading-extensions	heading-extensions	MDA	A value is generated from each value of the SET.
incomplete-copy	incomplete-copy	MD	The value of the parameter is the attribute value. If the source parameter is missing, an attribute with the default value shall be generated.
interchange-control-reference-for-this-recipient	interchange-control-reference	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.

TABLE 2 (continued)

Attribute-type-name	Source parameters by	Source generated	Generation rules
interchange-length	NONE	MS	The value is the number of octets occupied by the source parameter.
interchange-recipient-for-this-recipient	interchange-recipient	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
interchange-sender	interchange-sender	MD	The value of the parameter is the attribute value.
message-data	data	MD	The value of the parameter is the attribute value.
message-parameters	parameters	MD	The value of the parameter is the attribute value.
nn-extensions	nn-extensions	MDA	The value is generated from each value of the SET.
nn-reason-code	nn-reason-code	MD	The value of the parameter is the attribute value.
nn-supplementary-information	nn-supplementary-information	MD	The value of the parameter is the attribute value.
notification-security-elements	notification-security-elements	MD	The value of the parameter is the attribute value.
notification-time	notification-time	MD	The value of the parameter is the attribute value.
notifications-extensions	notifications-extensions	MDA	The value is generated from each value of the SET.
obsoleted-edims	obsoleted-EDIMs	MDA	The value is generated from each value of the SEQUENCE.
originator	originator	MD	The value of the parameter is the attribute value.
pn-extensions	pn-extensions	MDA	The value is generated from each value of the SET.
pn-supplementary-information	pn-supplementary-information	MDA	The value of the parameter is the attribute value.
processing-priority-code-for-this-recipient	processing-priority-code	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
recipient-extensions-for-this-recipient	recipient-extensions	MDA	The value is generated from each value of the SET in the recipient-sub-field for this recipient.
recipient-reference-for-this-recipient	recipient-reference	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.
related-messages	related-messages	MDA	The value is generated from each value of the SEQUENCE.

TABLE 2 (concluded)

Attribute-type-name	Source parameters by	Source generated	Generation rules
responsibility-forwarded	responsibility-forwarded	MD	The value of the parameter is the attribute value. If the source parameter is missing, an attribute with the default value shall be generated.
responsibility-passing-allowed-for-this-recipient	responsibility-passing-allowed	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient. If the source parameter is missing, an attribute with the default value shall be generated.
service-string-advice	service-string-advice	MD	The value of the parameter is the attribute value.
subject-edim	subject-edim	MD	The value of the parameter is the attribute value.
syntax-identifier	syntax-identifier	MD	The value of the parameter is the attribute value.
test-indicator-for-this-recipient	test-indicator	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient. If the source parameter is missing, an attribute with the default value shall be generated.
this-edim	this-EDIM	MD	The value of the parameter is the attribute value.
this-recipient	recipient	MD	The attribute-value is the value of the parameter in the recipient-sub-field for this recipient.

18.7.1 Summary Attributes

Some attributes summarize an EDI Messaging information object. These attributes are defined and described below.

18.7.1.1 EDIMS Entry Type

The `otEDIMS Entry Type` "ot:EDIMS Entry Type"§ attribute identifies an information object's type.

```
vaedims-entry-type "va:edims-entry-type"§ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIMSEntryType
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-sat-edims-entry-type
```

```
tyEDIMSEntryType "ty:EDIMSEntryType"§ ::= ENUMERATED {
  edim (0),
  pn (1),
  nn (2),
  fn (3) }
```

This attribute may assume any one of the following values:

- a) `edim`: The information object is an EDIM.
- b) `pn`: The information object is a PN.
- c) `nn`: The information object is an NN.
- d) `fn`: The information object is an FN.

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an EDIM or EDIN.

18.7.1.2 EDIM Synopsis

The `otEDIM Synopsis` `"ot:EDIM Synopsis"` attribute gives the structure, characteristics, size, and processing status of an EDIM at the granularity of individual body parts. This attribute is created when an EDIM is delivered to the MS.

```
vaedim-synopsisxe "va:edim-synopsis" $ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIMSynopsis
  SINGLE VALUE
  ::= id-sat-edim-synopsis
```

The synopsis of an EDIM comprises a synopsis of each of its body parts. The synopsis appear in the order in which the body parts appear.

```
tyEDIMSynopsisxe "ty:EDIMSynopsis" $ ::= SEQUENCE OF BodyPartSynopsis
```

The synopsis of a body part takes either of two forms depending upon whether the body part is of type Message or Non-message (i.e. body-parts other than a forwarded EDIM). This enables the synopsis of a forwarding EDIM to encompass the body parts of each forwarded EDIM (recursively), as well as those of the forwarding EDIM itself.

```
tyBodyPartSynopsisxe "ty:BodyPartSynopsis" $ ::= CHOICE {
  message                               [0] MessageBodyPartSynopsis,
  non-message                           [1] NonMessageBodyPartSynopsis }

tyMessageBodyPartSynopsisxe "ty:MessageBodyPartSynopsis" $ ::= SEQUENCE {
  number                               [0] SequenceNumber,
  synopsis                             [1] EDIMSynopsis }

tyNonMessageBodyPartSynopsisxe "ty:NonMessageBodyPartSynopsis" $ ::= SEQUENCE {
  type                                 [0] OBJECT IDENTIFIER,
  parameters                           [1] ExternallyDefinedParameters OPTIONAL,
  size [2] INTEGER,
  processed                            [3] BOOLEAN DEFAULT FALSE }
```

The synopsis of a Message body part has the following components:

- a) `otNumberxe "ot:Number"`: The sequence number that the MS assigns to the entry that the Message body part represents. This component is generated when a child-entry is created.
- b) `otSynopsisxe "ot:Synopsis"`: The synopsis of the EDIM that forms the content of the message that the body part represents. This component is generated when a child-entry is created.

The synopsis of a body part of type other than Message has the following components. For purposes of this synopsis, the body part is considered to be of type Externally Defined, whether or not it was so conveyed to the MS:

- c) otType "ot:Type": This value is generated when the entry is created. If the Non-message Body Part is an edi-body-part, the value is the object identifier value contained in the edi-bodypart-type attribute contained in this entry. If it is a removed-edi-body, the value is set to "id-syn-removed" (See Annex C). If it is a place-holder, the value is set to "id-syn-place-holder" (again, see Annex C). If it is an external-body-part, the value is set to the Direct-reference component of the body part's Data component.
- d) otParameters "ot:Parameters": This value is generated if the Non-message Body Part is an external-body-part. It contains that body part's Parameter component, which may describe the body part's format and control parameters.
- e) otSize "ot:Size": This value is created when the entry is created. The value is set to the size in octets of the encoding of the Encoding component of the body part's Data component when the Basic Encoding Rules of Recommendation X.209 are followed. If those rules permit several (e.g., both primitive and constructed) encodings of the component, the size may reflect any one of them.
- f) otProcessed "ot:Processed" (default false): An indication of whether or not the body part has been conveyed to the UA by means of the MS Fetch abstract operation, or has been processed by an auto action. This value is set to the default value when the EDIM is delivered to the MS and is updated as described in .

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an EDIM.

As a consequence of its variability, the value of the Size component should be considered only an estimate of the body part's size.

18.7.2 *EDI Notification Indicator*

The EDI Notification Indicator attribute contains information about whether any EDI Notifications have been sent in response to an EDIM, and if so which type of EDI Notifications were sent. The MS creates this attribute for each new EDIM and maintains the attribute values, depending on the auto actions performed.

```
vaedi-notification-indicator "va:edi-notification-indicator" $ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EDINotificationIndicator DEFAULT (0)
MATCHES FOR EQUALITY
MULTI VALUE ::=
id-sat-edi-notification-indicator
```

```
EDINotificationIndicator ::= ENUMERATED {
no-notification-sent (0),
pn-sent (1),
nn-sent (2),
fn-sent (3) }
```

Each value of this attribute may assume one of the following values:

- a) no-notification-sent: This is the initial value set by the MS when a new MS entry is created for the EDIM.
- b) pn-sent: This value means that the MS has generated and sent a Positive Notification (PN) in response to a request for a PN.
- c) nn-sent: This value means that the MS has generated and sent a Negative Notification (NN) in response to a request for an NN.
- d) fn-sent: This value means that the MS has generated and sent a Forwarded Notification (FN) in response to a request for an FN.

18.7.3 *Heading Attributes*

Some attributes are derived from the Heading of an EDIM. These attributes are defined and described below.

18.7.3.1 *Heading*

The otHeadingxe "ot:Heading"\$ attribute is the (entire) Heading of an EDIM.

```
vaheadingxe "va:heading"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX Heading
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-heading
```

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an EDIM.

18.7.3.2 *Heading fields*

Some attributes bear the names of Heading fields and have those fields as their values. Some attributes bear the names of Heading fields and have sub-fields of those fields as their values. See for semantics.

```
vathis-edimxe "va:this-edim"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ThisEDIMField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-this-edim
```

```
vaoriginatorxe "va:originator"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX OriginatorField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-originator
```

```
vaedin-receiverxe "va:edin-receiver"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINReceiverField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-hat-edin-receiver
```

varesponsibility-forwardedxe "va:responsibility-forwarded"\$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ResponsibilityForwarded
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-responsibility-forwarded

vaedi-bodypart-typexe "va:edi-bodypart-type"\$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EDIBodyPartType
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-edi-bodypart-type

vaincomplete-copyxe "va:incomplete-copy"\$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX IncompleteCopyField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-incomplete-copy

expiry-time; ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ExpiryTimeField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-expiry-time

va-related-messagesxe "va:related-messages"\$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX RelatedMessagesReference
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-related-messages

vaobsoleted-edimsxe "va:obsoleted-edims"\$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ObsoletedEDIMsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-obsoleted-edims

vaedi-application-security-elementxe "va:edi-application-security-element"\$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EDIApplicationSecurityElement
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-edi-application-security-element

vaedi-application-security-extensionsxe "va:edi-application-security-extensions"\$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EDIApplicationSecurityExtension
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-edi-application-security-extensions

vacross-referencing-informationxe "va:cross-referencing-information"\$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX CrossReferencingInformationSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-cross-referencing-information

Fields from EDIFACT Interchange:

vaedi-message-typexe "va:edi-message-type"\$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EDIMessageTypeFieldSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-edi-message-type

vaservice-string-advicexe "va:service-string-advice"\$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX ServiceStringAdviceField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-service-string-advice

vasyntax-identifierxe "va:syntax-identifier"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX SyntaxIdentifierField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-syntax-identifier

vainterchange-senderxe "va:interchange-sender"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX InterchangeSenderField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-interchange-sender

vadate-and-time-of-preparationxe "va:date-and-time-of-preparation"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX DateAndTimeOfPreparationField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-date-and-time-of-preparation

vaapplication-referencexe "va:application-reference"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX ApplicationReferenceField
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-hat-application-reference

Heading extensions:

vaheading-extensionsxe "va:heading-extensions"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX HeadingExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-heading-extensions

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an EDIM whose Heading contains the field or sub-field whose name the attribute bears.

18.7.3.3 *Recipient sub-field*

Some attributes bear the names of Recipient fields and have sub-fields of those fields as their values. See for semantics.

vathis-recipientxe "va:this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX RecipientField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-this-recipient

vaaction-request-for-this-recipientxe "va:action-request-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX ActionRequestField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-action-request-for-this-recipient

vaedi-notification-requests-for-this-recipientxe "va:edi-notification-requests-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX EDINotificationRequests
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-edi-notification-requests-for-this-recipient

vaedi-notification-security-for-this-recipientxe "va:edi-notification-security-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX EDINotificationSecurity
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-edi-notification-security-for-this-recipient

vaedi-reception-security-for-this-recipientxe "va:edi-reception-security-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX EDIReceptionSecurity
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-edi-reception-security-for-this-recipient

varesponsibility-passing-allowed-for-this-recipientxe "va:responsibility-passing-allowed-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX ResponsibilityPassingAllowedField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-responsibility-passing-allowed-for-this-recipient

-- Fields from EDIFACT interchange

vainterchange-recipient-for-this-recipientxe "va:interchange-recipient-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX InterchangeRecipientField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-interchange-recipient-for-this-recipient

varecipient-reference-for-this-recipientxe "va:recipient-reference-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX RecipientReferenceField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-recipient-reference-for-this-recipient

vainterchange-control-reference-for-this-recipientxe "va:interchange-control-reference-for-this-recipient"\$
ATTRIBUTE
WITH ATTRIBUTE-SYNTAX InterchangeControlReferenceField
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-rat-interchange-control-reference-for-this-recipient

vaprocessing-priority-code-for-this-recipientxe "va:processing-priority-code-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX ProcessingPriorityCodeField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-processing-priority-code-for-this-recipient

vaacknowledgement-request-for-this-recipientxe "va:acknowledgement-request-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX AcknowledgementRequestField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-acknowledgement-request-for-this-recipient

vacommunications-agreement-id-for-this-recipientxe "va:communications-agreement-id-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX CommunicationsAgreementIdField
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-rat-communications-agreement-id-for-this-recipient

vatest-indicator-for-this-recipientxe "va:test-indicator-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX TestIndicatorField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-test-indication-for-this-recipient

-- *END Fields from EDIFACT*

-- *Fields from ANSI X12 ISA*

```
vaauthorization-information-for-this-recipientxe "va:authorization-information-for-this-recipient"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX AuthorizationInformationField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-rat-authorization-information-for-this-recipient
```

-- *END Fields from ANSI X12 ISA*

Extensions:

```
varrecipient-extensions-for-this-recipientxe "va:recipient-extensions-for-this-recipient"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX RecipientExtensionsSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-rat-recipient-extensions-for-this-recipient
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an EDIM whose Heading Recipients field contains the field whose name the attribute bears. It shall maintain one attribute value for each sub-field.

18.7.4 *Body Attributes*

Some attributes are derived from the Body of an EDIM. These attributes are defined and described below.

18.7.4.1 *Body*

The otBodyxe "ot:Body"\$ attribute is the (entire) Body of an EDIM.

```
vabodyxe "va:body"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX Body
  SINGLE VALUE
  ::= id-bat-body
```

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an EDIM.

18.7.4.2 *Body Analyses*

Some attributes have as their values information about the body parts contained in the body of the message.

The interchange length attribute is created by the Message Store when it receives an EDIM. Its value indicates the length of the EDI Interchange carried in the Primary Body Part of the message.

```

va:interchange-lengthxe "va:interchange-length"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX InterchangeLength
  MATCHES FOR ORDERING
  SINGLE VALUE
  ::= id-bat-interchange-length

```

```

ty:InterchangeLengthxe "ty:InterchangeLength"$ ::= INTEGER

```

The Interchange Length gives the number of octets occupied by the EDI Interchange.

18.7.4.3 *Primary Body Parts*

Some attributes bear the names of the Primary Body Part types and have such body parts as their values. See for semantics.

```

va:edi-body-partxe "va:edi-body-part"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIBodyPart
  SINGLE VALUE
  ::= id-bat-edi-body-part

```

An MS holds each forwarded EDIM (i.e., each Message body part) as an information object in its own right, separate from the forwarding EDIM. (stored as a separate child entry in the stored-messages information base). That information object, of course, is a message whose content is an EDIM. The EDIM Body Parts attribute below, therefore, has as its values the sequence numbers the MS assigns to those messages. See for semantics.

```

va:edim-body-partxe "va:edim-body-part"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SequenceNumber -- sequence number of the forwarded EDIM entry.
  SINGLE VALUE
  ::= id-bat-edim-body-part

```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an EDIM. It shall maintain one attribute value for such a body part.

Some attributes bear the names the Parameters and Data components of an EDIM Body Part and have the Parameters and Data components as their values.

```

va:message-parametersxe "va:message-parameters"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX MessageParameters
  SINGLE VALUE
  ::= id-bat-message-parameters

```

```

va:message-dataxe "va:message-data"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX MessageData
  SINGLE VALUE
  ::= id-bat-message-data

```

An MS that supports these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an EDIM whose Body contains an EDIM Body Part.

18.7.4.4 *Externally Defined Body Part Types*

The `otExternally Defined Body Part Types` "ot:Externally Defined Body Part Types"§ attribute identifies the externally defined body part types represented in an EDIM.

```
vaexternally-defined-body-part-types "va:externally-defined-body-part-types"§ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
MATCHES FOR EQUALITY
MULTI VALUE
::= id-bat-externally-defined-body-part-types
```

An MS that supports this attribute shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an EDIM whose Body contains one or more Additional body parts. It shall maintain one attribute value for every such type present.

NOTE - Some Externally Defined body part types are defined in Recommendation X.420. Values are as specified in 7.3.12 of Recommendation X.420. Each value of this attribute corresponds to one of the attributes described below.

18.7.4.5 *Externally Defined Body Parts*

Some attributes, unnamed, have as their values the Encoding components (see) of the ASN.1 Externals that constitute the Data components of Externally Defined Body Parts.

To each externally defined body part type there correspond two attributes. The first attribute is denoted by the object identifier that is the Direct-reference component (again, see) of the External that constitutes the Data component of a body part of that type. The content of this first attribute contains the Body Part Reference and the Externally Defined Body Part encoded as an EDIM-ExternallyDefinedBodyPart (See).

The second attribute is denoted by the object identifier that is the Direct-reference component of the External that constitutes the Parameters component of a body part of that type. The content of this second attribute contains the Body Part Reference and the Parameters component encoded as follows:

```
tyEDIExternallyDefinedBodyPartParameterAttribute "ty:EDIExternallyDefinedBodyPartParameterAttribute"§ ::=
SEQUENCE {
    body-part-reference [0] BodyPartReference OPTIONAL,
    parameter [1] ExternallyDefinedParameters }
```

An MS that supports one of these body parts shall maintain both attributes for an information object that it holds if, and only if, that object is a message whose content is an EDIM whose Body contains one or more body parts of the type that corresponds to that attribute. It shall maintain one value of each attribute for each such body part.

NOTE - The externally defined body part attributes cannot be enumerated in practice because the externally defined body part types cannot be so enumerated.

The Externally Defined Body Part Types attribute determines the Externally Defined Body Part Types for a particular EDIM.

18.7.5 *Notification Attributes*

Some attributes are derived from an EDIN. These attributes are defined and described below.

18.7.5.1 *Common fields*

Some attributes bear the names of Common fields and have those fields as their values. See 6.1 for semantics.

```
vasubject-edimxe "va:subject-edim"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SubjectEDIMField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-subject-edim
```

```
vaedin-originatorxe "va:edin-originator"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINOriginatorField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-edin-originator
```

```
vafirst-recipientxe "va:first-recipient"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX FirstRecipientField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-first-recipient
```

```
vanotification-timexe "va:notification-time"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NotificationTimeField
  MATCHES FOR EQUALITY ORDERING
  SINGLE VALUE
  ::= id-nat-notification-time
```

```
vanotification-security-elementsxe "va:notification-security-elements"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SecurityElementsField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-notification-security-elements
```

```
vaedin-initiatorxe "va:edin-initiator"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDINInitiatorField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-edin-initiator
```

Some attributes bear the names of notification fields and have sub-fields of the Common fields of a notification as their values.

```
vanotification-extensionsex "va:notification-extensions"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NotificationExtensionsSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-nat-notification-extensions
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an EDIN that contains the field or sub-field whose name the attribute bears.

18.7.5.2 *Positive Notification fields*

Some attributes bear the names of PN EDIN fields and have those fields as their values. Some attributes bear the names of notification fields and have sub-fields of the PN fields of a notification as their values. See for semantics.

```
vapn-supplementary-informationxe "va:pn-supplementary-information"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDISupplementaryInformation
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-pn-supplementary-info
```

```
vapn-extensionsxe "va:pn-extensions"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX PNExtensionsSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-nat-pn-extensions
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is a PN contains the field whose name the attribute bears. It shall maintain one attribute value for each sub-field.

18.7.5.3 *Negative Notification fields*

Some attributes bear the names of NN EDIN fields and have those fields as their values. Some attributes bear the names of notification fields and have sub-fields of the NN fields of a notification as their values. See for semantics.

```
vann-reasonxe "va:nn-reason"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NNReasonCodeField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-nn-reason-code
```

```
vann-supplementary-informationxe "va:nn-supplementary-information"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDISupplementaryInformation
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-nn-supplementary-info
```

```
vann-extensionsxe "va:nn-extensions"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NNExtensionsSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-nat-nn-extensions
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an NN that contains the field whose name the attribute bears. It shall maintain one attribute value for each field or sub-field.

18.7.5.4 Forwarded Notification fields

Some attributes bear the names of FN EDIN fields and have those fields as their values. Some attributes bear the names of notification fields and have sub-fields of the FN fields of a notification as their values. See for semantics.

```
vafn-forwarded-toxe "va:forwarded-to" $ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ForwardedTo
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-forwarded-to
```

```
vafn-reason-codexe "va:fn-reason-code" $ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX FNReasonCodeField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-fn-reason-code
```

```
vafn-supplementary-informationxe "va:fn-supplementary-information" $ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDISupplementaryInformation
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-fn-supplementary-info
```

```
vafn-extensionsxe "va:fn-extensions" $ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX FNExtensionsSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-nat-fn-extensions
```

An MS that supports one of these attributes shall maintain it for an information object that it holds if, and only if, that object is a message whose content is an FN that contains the field whose name the attribute bears. It shall maintain one attribute value for each field or sub-field.

18.8 Procedures for EDI MS

The procedures for a general MS are specified in 14 and 15 of Recommendation X.413. This reference gives complementary information for MS systems that also explicitly support EDI messaging.

18.8.1 Additional procedures for message delivery

How the MS consumes the MTS abstract service is described in 14 of Recommendation X.413. The following text describes additional information about the procedures needed for EDI messaging.

If EDI Security Requests are present, then the EDI-auto-forward actions defined above may be prohibited, subject to the security policy in force. If EDI Security Requests are present then the EDI-auto-forward action (forwarding-with-responsibility-accepted) shall not be performed.

Addition to 14.1.1 bullet 2) a) of Recommendation X.413:

- If EDI auto forwarding criteria are registered by the Register-MS abstract operation, the new entry shall be matched against the criteria registered. The matching shall always proceed starting with the registration having the lowest registration identifier and perform the following auto actions:

- registrations against the "forward-with-responsibility-accepted" auto action.

If this results in forwarding being performed, it is possible that one or several forwardings may be performed for this EDIM.

- registrations against "forward-with-responsibility-not-accepted" auto-action.

If this results in a forwarding being performed, no further EDI forwarding actions shall be performed for this EDIM by the same EDI-MS.

If an auto action registration stipulates that the EDIM is to be deleted after forwarding, no further forwarding auto-action can take place.

The appropriate notification shall be returned for the first auto-forwarding that is performed for the EDIM.

When an EDIN is submitted, a value reflecting the type of the EDIN shall be added to the "edi-notification-indicator" attribute.

If an EDI auto-forwarding does not succeed, eg. through a non-delivery, an NN EDIN may be returned to the originator if an FN was previously sent.

19. Message Contents

As has already been seen, various secondary objects (e.g., UAs) have occasion to convey the Information Objects of as the contents of messages, as well as to convey probes concerning such messages. What follows specifies precisely how they shall do this.

The rules governing the transmittal of such messages and probes, and the semantics and abstract and transfer syntaxes of their contents, constitute glEDI Messagingx "gl:EDI Messaging"§.

19.1 Content

A secondary object that submits a message containing an EDIM or EDIN shall supply, as the octets of the Octet String that constitutes the content of the message, the result of encoding the Information Object of in accordance with the Basic Encoding Rules of Recommendation X.209.

19.2 Content Type

A secondary object that submits a message containing an EDIM or EDIN shall assign either the object identifier id-mct-pedi (see Annex A) or the integer value 35 to the Content Type. The object identifier shall be used if the MTA conforms to Recommendation X.411 in its 1988 version.

A secondary object that receives a message containing an EDIM or EDIN shall accept both the object identifier id-mct-pedi and the integer value 35 as Content Type.

Notwithstanding the provisions of Annex B of Recommendation X.419, an MTA that conforms to Recommendation X.411 in its 1988 version shall obey the following rules when it relays an MTS-APDU to an MTA that conforms to Recommendation X.411 in its 1984 version:

- If the Content Type in a message or probe is indicated by the integer value 35, it shall be unchanged.
- If the Content Type in a message containing an EDIM or EDIN is indicated by the object identifier id-mct-pedi, it shall be replaced by the integer value 35.

NOTE - When an MTA that conforms to Recommendation X.411 in its 1984 version relays an MTS-APDU to an MTA that conforms to Recommendation X.411 in its 1988 version, and the Content Type contains the integer value 35, the MTA that conforms to Recommendation X.411 in its 1988 version may replace the integer value 35 in the Content Type with the object identifier id-mct-pedi or leave the integer value unchanged.

19.3 Content Length

A secondary object that submits a probe concerning a message containing an EDIM or EDIN shall specify as the length of the message's content the size in octets of the encoding of the instance in question of the Information Object of (a choice of an EDIM or an EDIN) when the Basic Encoding Rules of Recommendation X.209 are followed. If those rules permit several (e.g., both primitive and constructed) encoding of that Information Object, the content length may reflect any one of them.

19.4 Encoded Information Types

A secondary object that submits a message containing an EDIM or EDIN shall specify the glEncoded Information Typesxe "gl:Encoded Information Types"§ (abEITxe "ab:EIT"§) of the message as follows.

In the case of an EDIN, the basic EITs shall be unspecified.

In the case of an EDIM, the EITs shall be the logical union of the EITs of the EDIM's body parts specified in accordance with the following rules:

- a) EDI Body Part: The EIT (if any) of the EDI Body Part shall have the same values as the Heading field EDI Body Part Type.
- b) EDIM Body Part(Forwarded Message): The EITs (if any) of a EDIM Body Part shall be those of the forwarded message.
- c) Additional body parts: The EIT of additional body parts (if any) shall be the logical union of the individual body parts EITs.

An Externally Defined body part whose extended type corresponds to a basic type shall be indicated using the built-in EIT.

The EDI Body Part Type may be indicated in the external EITs.

A secondary object that submits a message containing an EDIM to an MTA that conforms to Recommendation X.411 in its 1988 version shall use the union of the object identifiers from EDI Body Part Type (see and Annex A) for all "original-encoded-information-types".

A secondary object that submits a message containing an EDIM to an MTA that conforms to Recommendation X.411 in its 1984 version shall use the "undefined" bit of the "built-in-encoded-information-types" (called "basic-encoded-information-type" in Recommendation X.411 in its 1984 version), as no other indication is possible for the EITs defined in in an MTA that conforms to Recommendation X.411 in its 1984 version. The "external-encoded-information-type" field shall not be present.

NOTE - The following reduced functionality has to be considered when a secondary object submits a message containing an EDIM to an MTA that conforms to Recommendation X.411 in its 1984 version or when such messages are relayed through such an MTA. The delivering MTA cannot make a comparison of which EITs, and hence primary EDI body part types, the UA is prepared to accept for delivery (otherwise it would not perform the delivery at all). In addition, the security features of an MTA that conforms to Recommendation X.411 in its 1988 version cannot be used.

20. Port realization

How an MS or the MTS concretely realizes the secondary ports it supplies is specified in Recommendation X.419.

How a UA, TLMA, or AU concretely realizes the primary ports it supplies is beyond the scope of this Recommendation.

21. Conformance

The requirements a secondary object (excluding the MTS) and its implementor shall meet when the latter claims the former's conformance to this Recommendation are identified below. A number of the conformance requirements distinguish between support upon origination and support upon reception.

21.1 *Origination versus Reception*

A UA or AU shall be said to *gl:support upon origination* a particular Heading field, Heading extension, EDIM Body Part type or Externally Defined Body Part type if, and only if, it accepts, preserves, and emits, in full accord with this Recommendation, that particular Heading field or extension, or EDIM Body Part type or Externally Defined Body Part type, whenever a user calls upon it to convey an EDIM containing them to the MTS or the user's MS (the latter only in the case of a UA).

A UA or AU shall be said to *gl:support upon reception* a particular Heading field, Heading extension, EDIM Body Part type or Externally Defined Body Part type if, and only if, it accepts, preserves, and emits, in full accord with this Recommendation, that particular Heading field or extension, or EDIM Body Part type or Externally Defined Body Part type, whenever the MTS or a user's MS (the latter only in the case of a UA) calls upon it to convey to the user an EDIM containing them.

A PDAU supports nothing upon origination because it is not a supplier of the origination port.

21.2 *Statement requirements*

The implementor of a UA, MS or AU shall state the following. For each item below he shall make separate statements concerning conformance upon origination and conformance upon reception:

- a) The Heading fields for which he claims conformance.
- b) The body part types for which he claims conformance.
- c) In the case of a UA with MS or MS, the EDI Messaging-specific MS attributes for which it claims conformance.
- d) In the case of a UA with MS or MS, whether it supports the EDI messaging-specific auto actions.

21.3 *Static requirements*

A UA, MS or AU shall satisfy the following static requirements:

- a) A UA, MS or AU shall implement the Heading fields and the body part types for which conformance is claimed.
- b) A UA with MS or MS shall support the EDI messaging-specific MS attributes for which conformance is claimed, but including as a minimum those designated mandatory in . In addition, it shall support the mandatory attributes identified in Table 1 of Recommendation X.413.
- c) A UA, MS or AU shall concretely realize its abstract ports as specified in .
- d) A UA or MS shall be able to both submit and receive messages of the content types of . An AU shall be able to both import and export such messages.

21.4 *Dynamic requirements*

A UA, MS or AU shall satisfy the following dynamic requirements:

- a) A UA or MS shall follow the rules of operation specified in or , respectively.
- b) A UA, MS or AU shall submit and receive messages whose contents are as specified in .
- c) A UA, MS or AU shall register with the MTS its ability to accept delivery of messages of both of the content types of and EITs as specified in .

ANNEX A

Reference definition of Object Identifiers

(This annex forms an integral part of this Recommendation.)

The annex defines for reference purposes various Object Identifiers cited in the ASN.1 modules of subsequent annexes. It uses ASN.1.

All Object Identifiers this Recommendation assigns are assigned in this annex. The annex is definitive for all but those for ASN.1 modules, the object EDIMS application (EDIME) itself and the EDI use of Directories. The definitive assignments for the former occur in the modules themselves; other references to them appear in IMPORT statements. For the EDI use of Directories object identifiers, this annex only defines a base object identifier.

```

moEDIMSObjectIdentifiersxe "mo:EDIMSObjectIdentifiers" § {joint-iso-ccitt
  mhs-motis(6) edims(7) modules(0) object-identifiers(0) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

```

-- Prologue

-- Exports everything

```

IMPORTS -- nothing --;

tyIDxe "ty:ID" § ::= OBJECT IDENTIFIER

```

-- EDI Messaging (definitive)

```

idid-edimsxe "id:id-edims" § ID ::= { joint-iso-ccitt mhs-motis(6) edims(7) } -- This is definitive

```

-- Categories

idid-modxe "id:id-mod" §	ID ::= {id-edims 0} -- modules
idid-edixe "id:id-edi" §	ID ::= {id-edims 1} -- reserved
idid-otxe "id:id-ot" §	ID ::= {id-edims 2} -- object types
idid-ptxe "id:id-pt" §	ID ::= {id-edims 3} -- port types
idid-refxe "id:id-ref" §	ID ::= {id-edims 4} -- refinements
idid-satxe "id:id-sat" §	ID ::= {id-edims 5} -- summary attributes
idid-hatxe "id:id-hat" §	ID ::= {id-edims 6} -- heading attributes
idid-ratxe "id:id-rat" §	ID ::= {id-edims 7} -- recipient attributes
idid-batxe "id:id-bat" §	ID ::= {id-edims 8} -- body attributes
idid-natxe "id:id-nat" §	ID ::= {id-edims 9} -- notification attributes
idid-mctxe "id:id-mct" §	ID ::= {id-edims 10} -- message content types
idid-bpxe "id:id-bp" §	ID ::= {id-edims 11} -- edi body part types
idid-ntxe "id:id-nt" §	ID ::= {id-edims 12} -- edi notification types
idid-forxe "id:id-for" §	ID ::= {id-edims 13} -- edi action indicator types
idid-actxe "id:id-act" §	ID ::= {id-edims 14} -- edi auto-action identifier types
idid-dirxe "id:id-dir" §	ID ::= {id-edims 15} -- edi use of directories
idid-synxe "id:id-syn" §	ID ::= {id-edims 16} -- edi synopsis type

-- Modules

```

idid-mod-object-identifiersxe "id:id-mod-object-identifiers" § ID ::= {id-mod 0} -- definitive
idid-mod-functional-objectsxe "id:id-mod-functional-objects" § ID ::= {id-mod 1} -- definitive
idid-mod-information-objectsxe "id:id-mod-information-objects" § ID ::= {id-mod 2} -- definitive
idid-mod-abstract-servicexe "id:id-mod-abstract-service" § ID ::= {id-mod 3} -- definitive
idid-mod-message-store-attributesxe "id:id-mod-message-store-attributes" § ID ::= {id-mod 4} -- definitive
idid-mod-upper-boundsxe "id:id-mod-upper-bounds" § ID ::= {id-mod 5} -- definitive
idid-mod-edl-directory-cl-attxe "id:id-mod-edl-directory-cl-att" § ID ::= {id-mod 6} -- definitive
idid-mod-message-store-auto-actionsxe "id:id-mod-message-store-auto-actions" § ID ::= {id-mod 7} -- definitive

```

-- Object types

idid-ot-edime	xe "id:id-ot-edime"\$	ID ::= {id-ot 0}
idid-ot-edimg-user	xe "id:id-ot-edimg-user"\$	ID ::= {id-ot 1}
idid-ot-edims	xe "id:id-ot-edims"\$	ID ::= {id-ot 2}
idid-ot-edi-ua	xe "id:id-ot-edi-ua"\$	ID ::= {id-ot 3}
idid-ot-edi-ms	xe "id:id-ot-edi-ms"\$	ID ::= {id-ot 4}
idid-ot-pdau	xe "id:id-ot-pdau"\$	ID ::= {id-ot 5}

-- Port types

idid-pt-origination	xe "id:id-pt-origination"\$	ID ::= {id-pt 0}
idid-pt-reception	xe "id:id-pt-reception"\$	ID ::= {id-pt 1}

-- Refinements

idid-ref-primary	xe "id:id-ref-primary"\$	ID ::= {id-ref 0}
idid-ref-secondary	xe "id:id-ref-secondary"\$	ID ::= {id-ref 1}

-- EDI-Notification Types (for use in P1 notification extension field)

idid-nt-edi-pn	xe "id:id-nt-edi-pn"\$	ID ::= {id-nt 0}
idid-nt-edi-nn	xe "id:id-nt-edi-nn"\$	ID ::= {id-nt 1}
idid-nt-edi-fn	xe "id:id-nt-edi-fn"\$	ID ::= {id-nt 2}

-- Message content type

idid-mct-pedi	xe "id:id-mct-pedi"\$	ID ::= {id-mct 0} -- Pedi
---------------	-----------------------	---------------------------

-- EDI Body Part type (and P1 EIT)

idid-edifact-ISO646	xe "id:id-edifact-ISO646"\$	ID ::= {id-bp 0} -- ISO646 is equivalent to CCITT T.50
idid-edifact-T61	xe "id:id-edifact-T61"\$	ID ::= {id-bp 1}
idid-edifact-octet	xe "id:id-edifact-octet"\$	ID ::= {id-bp 2}
idid-ansiX12-ISO646	xe "id:id-ansiX12-ISO646"\$	ID ::= {id-bp 3}
idid-ansiX12-T61	xe "id:id-ansiX12-T61"\$	ID ::= {id-bp 4}
idid-ansiX12-octet	xe "id:id-ansiX12-octet"\$	ID ::= {id-bp 5}
idid-ansiX12-ebcdic	xe "id:id-ansiX12-ebcdic"\$	ID ::= {id-bp 6}
idid-untid-ISO646	xe "id:id-untid-ISO646"\$	ID ::= {id-bp 7}
idid-untid-T61	xe "id:id-untid-T61"\$	ID ::= {id-bp 8}
idid-untid-octet	xe "id:id-untid-octet"\$	ID ::= {id-bp 9}
idid-private-octet	xe "id:id-private-octet"\$	ID ::= {id-bp 10}
idid-undefined-octet	xe "id:id-undefined-octet"\$	ID ::= {id-bp 11}

-- EDI Action Indication

idid-for-action	xe "id:id-for-action"\$	ID ::= {id-for 0} -- For action
idid-for-copy	xe "id:id-for-copy"\$	ID ::= {id-for 1} -- copy, not original

-- EDIMG Specific Register Auto Actions

idid-act-edi-auto-forward	xe "id:id-act-edi-auto-forward"\$	ID ::= {id-act 0}
---------------------------	-----------------------------------	-------------------

-- *EDIM Synopsis (MS)*

```
idid-syn-removedxe "id:id-syn-removed"$ ID ::= {id-syn 0}
idid-syn-place-holderxe "id:id-syn-place-holder"$ ID ::= {id-syn 1}
```

-- *MESSAGE STORE ATTRIBUTES*

-- *Summary attributes*

```
idid-sat-edims-entry-typexe "id:id-sat-edims-entry-type"$ ID ::= {id-sat 0}
idid-sat-edim-synopsisxe "id:id-sat-edim-synopsis"$ ID ::= {id-sat 1}
idid-sat-edi-notification-indicatorxe "id:id-sat-edi-notification-indicator"$ ID ::= {id-sat 2}
```

-- *Heading attributes*

```
idid-hat-headingxe "id:id-hat-heading"$ ID ::= {id-hat 0}
idid-hat-this-edimxe "id:id-hat-this-edim"$ ID ::= {id-hat 1}
idid-hat-originatorxe "id:id-hat-originator"$ ID ::= {id-hat 2}
idid-hat-recipientsxe "id:id-hat-recipients"$ ID ::= {id-hat 3}
idid-hat-edin-receiverxe "id:id-hat-edin-receiver"$ ID ::= {id-hat 4}
idid-hat-responsibility-forwardedxe "id:id-hat-responsibility-forwarded"$ ID ::= {id-hat 5}
idid-hat-edi-bodypart-typexe "id:id-hat-edi-bodypart-type"$ ID ::= {id-hat 6}
idid-hat-incomplete-copyxe "id:id-hat-incomplete-copy"$ ID ::= {id-hat 7}
idid-hat-expiry-timexe "id:id-hat-expiry-time"$ ID ::= {id-hat 8}
idid-hat-related-messagesxe "id:id-hat-related-messages"$ ID ::= {id-hat 9}
idid-hat-obsolete-edimsxe "id:id-hat-obsolete-edims"$ ID ::= {id-hat 10}
idid-hat-edi-application-security-elementxe "id:id-hat-edi-application-security-element"$ ID ::= {id-hat 11}
idid-hat-edi-application-security-extensionsexxe "id:id-hat-edi-application-security-extensionsexxe"$ ID ::= {id-hat 12}
idid-hat-cross-referencing-informationxe "id:id-hat-cross-referencing-information"$ ID ::= {id-hat 13}
idid-hat-edi-message-typexe "id:id-hat-edi-message-type"$ ID ::= {id-hat 14}
idid-hat-service-string-advicexe "id:id-hat-service-string-advice"$ ID ::= {id-hat 15}
idid-hat-syntax-identifierxe "id:id-hat-syntax-identifier"$ ID ::= {id-hat 16}
idid-hat-interchange-senderxe "id:id-hat-interchange-sender"$ ID ::= {id-hat 17}
idid-hat-date-and-time-of-preparationxe "id:id-hat-date-and-time-of-preparation"$ ID ::= {id-hat 18}
idid-hat-application-referencexe "id:id-hat-application-reference"$ ID ::= {id-hat 19}
idid-hat-heading-extensionsexxe "id:id-hat-heading-extensionsexxe"$ ID ::= {id-hat 20}
```

-- *Per Recipient attributes*

```

idid-rat-this-recipientxe "id:id-rat-this-recipient"$ ID ::= {id-rat 10}
idid-rat-action-request-for-this-recipientxe "id:id-rat-action-request-for-this-recipient"$ ID ::= {id-rat 1}
idid-rat-edi-notification-requests-for-this-recipientxe "id:id-rat-edi-notification-requests-for-this-recipient"$
ID ::= {id-rat 2}
idid-rat-responsibility-passing-allowed-for-this-recipientxe "id:id-rat-responsibility-passing-allowed-for-this-
recipient"$ ID ::= {id-rat 3}
-- UNB EDIFACT Field Object Ids --
idid-rat-interchange-recipient-for-this-recipientxe "id:id-rat-interchange-recipient-for-this-recipient"$
ID ::= {id-rat 4}
idid-rat-recipient-reference-for-this-recipientxe "id:id-rat-recipient-reference-for-this-recipient"$
ID ::= {id-rat 5}
idid-rat-interchange-control-reference-for-this-recipientxe "id:id-rat-interchange-control-reference-for-this-
recipient"$ ID ::= {id-rat 6}
idid-rat-processing-priority-code-for-this-recipientxe "id:id-rat-processing-priority-code-for-this-recipient"$
ID ::= {id-rat 7}
idid-rat-acknowledgement-request-for-this-recipientxe "id:id-rat-acknowledgement-request-for-this-
recipient"$ ID ::= {id-rat 8}
idid-rat-communications-agreement-id-for-this-recipientxe "id:id-rat-communications-agreement-id-for-this-
recipient"$ ID ::= {id-rat 9}
idid-rat-test-indicator-for-this-recipientxe "id:id-rat-test-indicator-for-this-recipient"$ ID ::= {id-rat 10}
idid-rat-notification-security-for-this-recipientxe "id:id-rat-notification-security-for-this-recipient"$
ID ::= {id-rat 11}
idid-rat-edi-reception-security-for-this-recipientxe "id:id-rat-edi-reception-security-for-this-recipient"$
ID ::= {id-rat 12}
idid-rat-recipient-extensions-for-this-recipientxe "id:id-rat-recipient-extensions-for-this-recipient"$
ID ::= {id-rat 13}
-- ANSI X12 ISA Field Object Ids --
.I.id:id-rat-authorization-information-for-this-recipient; ID ::= {id-rat 14}

```

-- Body attributes

```

idid-bat-bodyxe "id:id-bat-body"$ ID ::= {id-bat 0}
idid-bat-interchange-lengthxe "id:id-bat-interchange-length"$ ID ::= {id-bat 1}
idid-bat-edi-body-partxe "id:id-bat-edi-body-part"$ ID ::= {id-bat 2}
idid-bat-edim-body-partxe "id:id-bat-edim-body-part"$ ID ::= {id-bat 3}
idid-bat-message-parametersxe "id:id-bat-message-parameters"$ ID ::= {id-bat 4}
idid-bat-message-dataxe "id:id-bat-message-data"$ ID ::= {id-bat 5}
idid-bat-externally-defined-body-part-typesxe "id:id-bat-externally-defined-body-part-types"$ ID ::= {id-bat 6}

```

-- Notification attributes

```

idid-nat-subject-edimxe "id:id-nat-subject-edim"$ ID ::= {id-nat 0}
idid-nat-edin-originatorxe "id:id-nat-edin-originator"$ ID ::= {id-nat 1}
idid-nat-first-recipientxe "id:id-nat-first-recipient"$ ID ::= {id-nat 2}
idid-nat-notification-timexxe "id:id-nat-notification-time"$ ID ::= {id-nat 3}
idid-nat-notification-security-elementsxe "id:id-nat-notification-security-elements"$ ID ::= {id-nat 4}
idid-nat-notification-extensionsxe "id:id-nat-notification-extensions"$ ID ::= {id-nat 5}
idid-nat-edin-initiatorxe "id:id-nat-edin-initiator"$ ID ::= {id-nat 6}
-- PN attributes
idid-nat-pn-supplementary-infoxe "id:id-nat-pn-supplementary-info"$ ID ::= {id-nat 7}
idid-nat-pn-extensionsxe "id:id-nat-pn-extensions"$ ID ::= {id-nat 8}
-- NN attributes
idid-nat-nn-reason-codexxe "id:id-nat-nn-reason-code"$ ID ::= {id-nat 9}
idid-nat-nn-supplementary-infoxe "id:id-nat-nn-supplementary-info"$ ID ::= {id-nat 10}
idid-nat-nn-extensionsxe "id:id-nat-nn-extensions"$ ID ::= {id-nat 11}
-- FN attributes
idid-nat-forwarded-toxe "id:id-nat-forwarded-to"$ ID ::= {id-nat 12}
idid-nat-fn-reason-codexxe "id:id-nat-fn-reason-code"$ ID ::= {id-nat 13}
idid-nat-fn-supplementary-infoxe "id:id-nat-fn-supplementary-info"$ ID ::= {id-nat 14}
idid-nat-fn-extensionsxe "id:id-nat-fn-extensions"$ ID ::= {id-nat 15}

```

-- MESSAGE STORE ATTRIBUTES - END

END -- of EDIMSObjectIdentifiers

ANNEX B

Reference definition of Abstract Information Objects

(This annex forms an integral part of this Recommendation.)

This annex defines for reference purposes the abstract information objects of EDI Messaging. It defines a Body Part for EDIM that includes a body part reference number while importing the IPMS externally defined MACRO for specifying non-EDI body parts. It also defines an EDIM-EXTENSION MACRO that differs from IPMS.

```
mo EDIMSInformationObjectsxe "mo: EDIMSInformationObjects"$ {joint-iso-ccitt
    mhs-motis(6) edims(7) modules(0) information-objects(2) }
```

```
DEFINITIONS IMPLICIT TAGS ::=
BEGIN
```

-- Prologue

-- Exports everything

```
IMPORTS
```

-- EDIMS Upper bounds

```
ub-application-reference, ub-authorization-information,
ub-authorization-information-qualifier, ub-communications-agreement-id,
ub-edi-application-security-elements, ub-edi-message-type,
ub-identification-code, ub-identification-code-qualifier, ub-interchange-control-reference,
ub-local-reference, ub-processing-priority-code, ub-reason-code, ub-recipient-reference,
ub-recipient-reference-qualifier, ub-routing-address, ub-syntax-identifier,
ub-syntax-version
----
FROM EDIMSupperBounds {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) upper-bounds(5) }
```

-- EDIMS Object Identifiers

```
id-edifact-ISO646, id-for-action
-----
FROM EDIMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) object-identifiers(0) }
```

-- MTS Upper Bounds

```
ub-bit-options, ub-integer-options, ub-supplementary-info-length
-----
FROM MTSUpperBounds {joint-iso-ccitt mhs-motis(6) mts(3) modules(0) upper-bounds(3) }
```

-- *MTS Abstract Service*

```
MessageDeliveryTime, ORAddress, ORName, OtherMessageDeliveryFields, ContentIntegrityCheck
----
FROM MTSAbstractService {joint-iso-ccitt mhs-motis(6) mts(3) modules(0) mts-abstract-service(1) }
```

-- *IPM Abstract Service*

```
EXTENDED-BODY-PART-TYPE, ExternallyDefinedBodyPart,
----
FROM IPMSInformationObjects {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) information-objects(2) };
```

-- *END Imports*

-- *ABSTRACT INFORMATION OBJECTS*

-- *Overview*

```
tyInformationObjectxe "ty:InformationObject"$ ::= CHOICE {
  edim          [0] EDIM,
  edin          [1] EDIN }
```

-- *Common data types*

-- *EDIM Identifier*

```
tyEDIMIdentifierxe "ty:EDIMIdentifier"$ ::= SET {
  user          [0] ORName,
  user-relative-identifier [1] LocalReference }

tyLocalReferencexe "ty:LocalReference"$ ::= PrintableString (SIZE (0..ub-local-reference))
```

-- *Extensions*

```
tyExtensionFieldxe "ty:ExtensionField"$ ::= SEQUENCE {
  type          [0] EDIM-EXTENSION,
  criticality    [1] Criticality DEFAULT FALSE,
  value          [2] ANY DEFINED BY type DEFAULT NULL NULL }
```

```
tyCriticalityxe "ty:Criticality"$ ::= BOOLEAN
```

-- *EDIM Extension MACRO*

```
maEDIM-EXTENSIONxe "ma:EDIM-EXTENSION"$ MACRO ::=
  BEGIN

  TYPE NOTATION          ::= DataType Critical | empty
  VALUE NOTATION          ::= value(VALUE OBJECT IDENTIFIER)

  DataType                ::= type (X) Default | empty
  Default                  ::= "DEFAULT" value (X) | empty
  Critical                  ::= "CRITICAL" | empty

  END -- of extension
```

-- EDI Messages

```
tyEDIMxe "ty:EDIM"$ ::= SEQUENCE {  
    heading      Heading,  
    body         Body }
```

-- Heading Field Component Types

-- Interchange Recipient / Sender

-- Identification Code

```
tyIdentificationCodexe "ty:IdentificationCode"$ ::= T61String (SIZE (1..ub-identification-code))
```

-- Identification Code Qualifier

```
tyIdentificationCodeQualifierxe "ty:IdentificationCodeQualifier"$ ::= T61String (SIZE (1..ub-identification-code-  
qualifier))
```

-- Routing Address

```
tyRoutingAddressxe "ty:RoutingAddress"$ ::= T61String (SIZE (1..ub-routing-address))
```

-- Heading Fields

```
tyHeadingxe "ty:Heading"$ ::= SEQUENCE {  
    this-EDIM      [1] ThisEDIMField,  
    originator     [2] OriginatorField OPTIONAL,  
    recipients     [3] RecipientsField OPTIONAL,  
    edin-receiver  [4] EDINReceiverField OPTIONAL,  
    responsibility-forwarded [5] ResponsibilityForwarded DEFAULT FALSE,  
    edi-bodypart-type [6] EDIBodypartType DEFAULT {id-edifact-ISO646},  
    incomplete-copy [7] IncompleteCopyField DEFAULT FALSE,  
    expiry-time    [8] ExpiryTimeField OPTIONAL,  
    related-messages [9] RelatedMessagesField OPTIONAL,  
    obsoleted-EDIMs [10] ObsoletedEDIMsField OPTIONAL,  
    edi-application-security-elements [11] EDIApplicationSecurityElementsField OPTIONAL,  
    cross-referencing-information [12] CrossReferencingInformationField OPTIONAL,  
  
    -- Begin Fields from EDIFACT Interchange  
    edi-message-type [13] EDIMessageTypeField OPTIONAL,  
    service-string-advice [14] ServiceStringAdviceField OPTIONAL,  
    syntax-identifier [15] SyntaxIdentifierField OPTIONAL,  
    interchange-sender [16] InterchangeSenderField OPTIONAL,  
    date-and-time-of-preparation [17] DateAndTimeOfPreparationField OPTIONAL,  
    application-reference [18] ApplicationReferenceField OPTIONAL,  
    -- End Fields from EDIFACT  
  
    heading-extensions [19] HeadingExtensionsField OPTIONAL }
```

-- This EDIM

```
tyThisEDIMFieldxe "ty:ThisEDIMField"$ ::= EDIMIdentifier
```

-- Originator

```
tyOriginatorFieldxe "ty:OriginatorField"$ ::= ORName
```

-- Recipients

tyRecipientsFieldxe "ty:RecipientsField"\$::= SET OF RecipientsSubField

tyRecipientsSubFieldxe "ty:RecipientsSubField"\$::= SEQUENCE {
 recipient [1] RecipientField,
 action-request [2] ActionRequestField DEFAULT {id-for-action},
 edi-notification-requests-field [3] EDINotificationRequestsField OPTIONAL,
 responsibility-passing-allowed [4] ResponsibilityPassingAllowedField DEFAULT FALSE,

 -- Begin Fields from EDIFACT UNB
 interchange-recipient [5] InterchangeRecipientField OPTIONAL,
 recipient-reference [6] RecipientReferenceField OPTIONAL,
 interchange-control-reference [7] InterchangeControlReferenceField OPTIONAL,
 processing-priority-code [8] ProcessingPriorityCodeField OPTIONAL,
 acknowledgement-request [9] AcknowledgementRequestField DEFAULT FALSE,
 communications-agreement-id [10] CommunicationsAgreementIdField OPTIONAL,
 test-indicator [11] TestIndicatorField DEFAULT FALSE,
 -- End Fields from EDIFACT UNB

 -- Begin Fields from ANSI X12 ISA
 authorization-information [12] AuthorizationInformationField OPTIONAL,
 -- End Fields from ANSI X12 ISA

 recipient-extensions [13] RecipientExtensionsField OPTIONAL }

-- Recipient

tyRecipientFieldxe "ty:RecipientField"\$::= ORName

-- Action Request

tyActionRequestFieldxe "ty:ActionRequestField"\$::= OBJECT IDENTIFIER

-- EDI Notification Requests

tyEDINotificationRequestsFieldxe "ty:EDINotificationRequestsField"\$::= SEQUENCE {
 edi-notification-requests [0] EDINotificationRequests DEFAULT {},
 edi-notification-security [1] EDINotificationSecurity DEFAULT {},
 edi-reception-security [2] EDIReceptionSecurity DEFAULT {} }

tyEDINotificationRequestsxe "ty:EDINotificationRequests"\$::= BIT STRING {
 pn (0),
 nn (1),
 fn (2) }(SIZE (0..ub-bit-options))

tyEDINotificationSecurityxe "ty:EDINotificationSecurity"\$::= BIT STRING {
 proof (0),
 non-repudiation (1) }(SIZE (0..ub-bit-options))

tyEDIReceptionSecurityxe "ty:EDIReceptionSecurity"\$::= BIT STRING {
 proof (0),
 non-repudiation (1) }(SIZE (0..ub-bit-options))

-- Interchange recipient

tyInterchangeRecipientFieldxe "ty:InterchangeRecipientField"\$::= SEQUENCE {
 recipient-identification-code [0] IdentificationCode,
 identification-code-qualifier [1] IdentificationCodeQualifier OPTIONAL,
 routing-address [2] RoutingAddress OPTIONAL }

-- Recipient reference

```
tyRecipientReferenceFieldxe "ty:RecipientReferenceField"$ ::= SEQUENCE {  
    recipient-reference [0] RecipientReference,  
    recipient-reference-qualifier [1] RecipientReferenceQualifier OPTIONAL }  
  
tyRecipientReferencexe "ty:RecipientReference"$ ::= T61String (SIZE (1..ub-recipient-reference))  
  
tyRecipientReferenceQualifierxe "ty:RecipientReferenceQualifier"$ ::= T61String (SIZE (1..ub-recipient-reference-qualifier))
```

-- Recipient Extensions

```
tyRecipientExtensionsFieldxe "ty:RecipientExtensionsField"$ ::= SET OF RecipientExtensionsSubField  
  
tyRecipientExtensionsSubFieldxe "ty:RecipientExtensionsSubField"$ ::= ExtensionField
```

-- EDIN receiver

```
tyEDINReceiverFieldxe "ty:EDINReceiverField"$ ::= SEQUENCE {  
    edin-receiver-name [0] ORName,  
    original-edim-identifier [1] EDIMIdentifier OPTIONAL,  
    first-recipient [2] FirstRecipientField OPTIONAL}
```

-- Responsibility Forwarded indication

```
tyResponsibilityForwardedxe "ty:ResponsibilityForwarded"$ ::= BOOLEAN -- Default False
```

-- EDI Body Part Types - identifies EDI Standard, Character set and encoding

```
tyEDIBodyPartTypexe "ty:EDIBodyPartType"$ ::= OBJECT IDENTIFIER -- default EDIFACT-ISO646
```

-- EDI message type

```
tyEDIMessageTypeFieldxe "ty:EDIMessageTypeField"$ ::= SET OF EDIMessageTypeFieldSubField  
  
tyEDIMessageTypeFieldSubFieldxe "ty:EDIMessageTypeFieldSubField"$ ::= T61String (SIZE (1..ub-edimessage-type))
```

-- Responsibility Passing Allowed

```
tyResponsibilityPassingAllowedFieldxe "ty:ResponsibilityPassingAllowedField"$ ::= BOOLEAN -- Default FALSE
```

-- Incomplete Copy

```
tyIncompleteCopyFieldxe "ty:IncompleteCopyField"$ ::= BOOLEAN -- Default False
```

-- Expiry time

```
tyExpiryTimeFieldxe "ty:ExpiryTimeField"$ ::= UTCTime
```

-- Related Messages

tyRelatedMessagesFieldxe "ty:RelatedMessagesField"\$::= SEQUENCE OF RelatedMessageReference

tyRelatedMessageReferencexxe "ty:RelatedMessageReference"\$::= CHOICE {
edi-message-reference [0] EDIMIdentifier,
external-message-reference [1] ExternalMessageReference }

tyExternalMessageReferencexxe "ty:ExternalMessageReference"\$::= EXTERNAL

-- Obsoleted EDIMs

tyObsoletedEDIMsFieldxe "ty:ObsoletedEDIMsField"\$::= SEQUENCE OF ObsoletedEDIMsSubfield

tyObsoletedEDIMsSubfieldxe "ty:ObsoletedEDIMsSubfield"\$::= EDIMIdentifier

-- EDI Application Security Elements

tyEDIApplicationSecurityElementsFieldxe "ty:EDIApplicationSecurityElementsField"\$::= SEQUENCE {
edi-application-security-element [0] EDIApplicationSecurityElement OPTIONAL,
edi-encrypted-primary-bodypart [1] BOOLEAN OPTIONAL,
edi-application-security-extensions [2] EDIApplicationSecurityExtensions OPTIONAL }

tyEDIApplicationSecurityElementxe "ty:EDIApplicationSecurityElement"\$::= BIT STRING (SIZE (0..ub-edi-application-security-elements))

tyEDIApplicationSecurityExtensionsexxe "ty:EDIApplicationSecurityExtensions"\$::= SEQUENCE OF EDIApplicationSecurityExtension

tyEDIApplicationSecurityExtensionxe "ty:EDIApplicationSecurityExtension"\$::= ExtensionsField

-- Cross Referencing Information

tyCrossReferencingInformationFieldxe "ty:CrossReferencingInformationField"\$::= SET OF CrossReferencingInformationSubField

tyCrossReferencingInformationSubFieldxe "ty:CrossReferencingInformationSubField"\$::= SEQUENCE {
application-cross-reference [0] ApplicationCrossReference,
message-reference [1] MessageReference OPTIONAL,
body-part-reference [2] BodyPartReference }

tyApplicationCrossReferencexxe "ty:ApplicationCrossReference"\$::= OCTET STRING

tyMessageReferencexxe "ty:MessageReference"\$::= EDIMIdentifier

-- Service String Advice

tyServiceStringAdviceFieldxe "ty:ServiceStringAdviceField"\$::= SEQUENCE {
component-data-element-separator [0] ComponentDataElementSeparator,
data-element-separator [1] DataElementSeparator,
decimal-notation [2] DecimalNotation,
release-indicator [3] ReleaseIndicator OPTIONAL,
reserved [4] Reserved OPTIONAL,
segment-terminator [5] SegmentTerminator }

tyComponentDataElementSeparatorxe "ty:ComponentDataElementSeparator"\$::= OCTET STRING (SIZE (1))

tyDataElementSeparatorxe "ty:DataElementSeparator"\$::= OCTET STRING (SIZE (1))

```

tyDecimalNotationxe "ty:DecimalNotation"$ ::= OCTET STRING (SIZE (1))

tyReleaseIndicatorxe "ty:ReleaseIndicator"$ ::= OCTET STRING (SIZE (1))

tyReservedxe "ty:Reserved"$          ::= OCTET STRING (SIZE (1))

tySegmentTerminatorxe "ty:SegmentTerminator"$      ::= OCTET STRING (SIZE (1))

```

-- Syntax Identifier

```

tySyntaxIdentifierFieldxe "ty:SyntaxIdentifierField"$ ::= SEQUENCE {
    syntax-identifier      SyntaxIdentifier,
    syntax-version         SyntaxVersion }

tySyntaxIdentifierxe "ty:SyntaxIdentifier"$ ::= T61String (SIZE (1..ub-syntax-identifier))

tySyntaxVersionxe "ty:SyntaxVersion"$ ::= PrintableString (SIZE (1..ub-syntax-version))

```

-- Interchange sender

```

tyInterchangeSenderFieldxe "ty:InterchangeSenderField"$ ::= SEQUENCE {
    sender-identification      [0] IdentificationCode,
    identification-code-qualifier [1] IdentificationCodeQualifier OPTIONAL,
    address-for-reverse-routing [2] RoutingAddress OPTIONAL } -- EDIFACT Routing Information

```

-- Date and Time of preparation

```

tyDateAndTimeOfPreparationFieldxe "ty:DateAndTimeOfPreparationField"$ ::= UTCTime

```

-- Interchange control reference

```

tyInterchangeControlReferenceFieldxe "ty:InterchangeControlReferenceField"$ ::= T61String (SIZE (1..ub-
interchange-control-reference))

```

-- Application reference

```

tyApplicationReferenceFieldxe "ty:ApplicationReferenceField"$ ::= T61String (SIZE (1..ub-application-reference))

```

-- Processing Priority Code

```

tyProcessingPriorityCodeFieldxe "ty:ProcessingPriorityCodeField"$ ::= T61String (SIZE (1..ub-processing-
priority-code))

```

-- Acknowledgement Request

```

tyAcknowledgementRequestFieldxe "ty:AcknowledgementRequestField"$ ::= BOOLEAN -- default FALSE

```

-- Communications Agreement Id

```

tyCommunicationsAgreementIdFieldxe "ty:CommunicationsAgreementIdField"$ ::= T61String (SIZE (1..ub-
communications-agreement-id))

```

-- Test indicator

```

tyTestIndicatorFieldxe "ty:TestIndicatorField"$ ::= BOOLEAN -- default FALSE

```

-- Authorization Information

```
tyAuthorizationInformationFieldxe "ty:AuthorizationInformationField"$ ::= SEQUENCE {
    authorization-information          [0] AuthorizationInformation,
    authorization-information-qualifier [1] AuthorizationInformationQualifier OPTIONAL }

tyAuthorizationInformationxe "ty:AuthorizationInformation"$ ::= T61String (SIZE (1..ub-authorization-
information))

tyAuthorizationInformationQualifierxe "ty:AuthorizationInformationQualifier"$ ::= T61String (SIZE (1..ub-
authorization-information-qualifier))
```

-- Heading Extensions

```
tyHeadingExtensionsFieldxe "ty:HeadingExtensionsField"$ ::= SET OF HeadingExtensionsSubField

tyHeadingExtensionsSubFieldxe "ty:HeadingExtensionsSubField"$ ::= ExtensionField
```

-- EDIM body

```
tyBodyxe "ty:Body"$ ::= SEQUENCE {
    primary-body-part          PrimaryBodyPart,
    additional-body-parts      OtherBodyParts OPTIONAL }

tyPrimaryBodyPartxe "ty:PrimaryBodyPart"$ ::= CHOICE {
    edi-body-part              [0] EDIBodyPart,
    forwarded-EDIM             [1] EDIMBodyPart }

tyOtherBodyPartsex "ty:OtherBodyParts"$ ::= SEQUENCE OF EDIM-ExternallyDefinedBodyPart
```

-- EDI body part

```
tyEDIBodyPartxe "ty:EDIBodyPart"$ ::= OCTET STRING
```

-- Forwarded EDIM body part

```
tyEDIMBodyPartxe "ty:EDIMBodyPart"$ ::= SEQUENCE {
    parameters          [0] MessageParameters OPTIONAL,
    data                 [1] MessageData }

tyMessageParametersxe "ty:MessageParameters"$ ::= SET {
    delivery-time          [0] MessageDeliveryTime OPTIONAL,
    delivery-envelope      [1] OtherMessageDeliveryFields OPTIONAL,
    other-parameters       [2] EDISupplementaryInformation OPTIONAL }

tyMessageDataxe "ty:MessageData"$ ::= SEQUENCE {
    heading                Heading,
    body                   BodyOrRemoved }

tyBodyOrRemovedxe "ty:BodyOrRemoved"$ ::= SEQUENCE {
    primary-or-removed      PrimaryOrRemoved,
    additional-body-parts   AdditionalBodyParts OPTIONAL }

tyPrimaryOrRemovedxe "ty:PrimaryOrRemoved"$ ::= CHOICE {
    removed-edi-body        [0] NULL,
    primary-body-part        [1] EXPLICIT PrimaryBodyPart }

tyAdditionalBodyPartsex "ty:AdditionalBodyParts"$ ::= SEQUENCE OF CHOICE {
    external-body-part      [0] EDIM-ExternallyDefinedBodyPart,
    place-holder             [1] BodyPartPlaceHolder } -- This type is for Body Part Removal
```

```

tyBodyPartPlaceHolderxe "ty:BodyPartPlaceHolder"$ ::= EDIM-ExternallyDefinedBodyPart -- Only the data
-- portion of the Externally Defined Body shall be removed.
-- See text in .

```

-- EDIM Externally Defined Body Parts

```

tyEDIM-ExternallyDefinedBodyPartxe "ty:EDIM-ExternallyDefinedBodyPart"$ ::= SEQUENCE {
    body-part-reference          [0] BodyPartReference OPTIONAL,
    external-body-part          [1] ExternallyDefinedBodyPart -- from IPMS --}

```

```

tyBodyPartReferencexe "ty:BodyPartReference"$ ::= INTEGER -- shall be unique within a EDIM

```

-- Supplementary Info

```

tyEDISupplementaryInformationxe "ty:EDISupplementaryInformation"$ ::= IA5String (SIZE (1..ub-
supplementary-info-length))

```

-- EDI Notifications (EDINs)

```

tyEDINxe "ty:EDIN"$ ::= CHOICE {
    positive-notification      [0] PositiveNotificationFields,
    negative-notification      [1] NegativeNotificationFields,
    forwarded-notification     [2] ForwardedNotificationFields }

```

```

tyPNxe "ty:PN"$ ::= EDIN -- with positive-notification chosen

```

```

tyNNxe "ty:NN"$ ::= EDIN -- with negative-notification chosen

```

```

tyFNxe "ty:FN"$ ::= EDIN -- with forwarded-notification chosen

```

-- Common fields

```

tyCommonFieldsxe "ty:CommonFields"$ ::= SEQUENCE {
    subject-edim               [1] SubjectEDIMField,
    edin-originator            [2] EDINOriginatorField,
    first-recipient            [3] FirstRecipientField OPTIONAL,
    notification-time          [4] NotificationTimeField,
    notification-security-elements [5] SecurityElementsField OPTIONAL,
    edin-initiator             [6] EDINInitiatorField,
    notifications-extensions   [7] NotificationExtensionsField OPTIONAL }

```

-- Subject EDIM Identifier

```

tySubjectEDIMFieldxe "ty:SubjectEDIMField"$ ::= EDIMIdentifier

```

-- EDI Notification Originator

```

tyEDINOriginatorFieldxe "ty:EDINOriginatorField"$ ::= ORName

```

-- First Recipient

```

tyFirstRecipientFieldxe "ty:FirstRecipientField"$ ::= ORName

```

-- Notification Time

```

tyNotificationTimeFieldxe "ty:NotificationTimeField"$ ::= UTCTime

```

-- Security Elements

```
tySecurityElementsFieldxe "ty:SecurityElementsField"$ ::= SEQUENCE {  
    original-content [0] Content OPTIONAL,  
    original-content-integrity-check [1] ContentIntegrityCheck OPTIONAL,  
    edi-application-security-elements [2] EDIApplicationSecurityElementsField OPTIONAL,  
    security-extensions [3] SecurityExtensionsField OPTIONAL }
```

```
tySecurityExtensionsFieldxe "ty:SecurityExtensionsField"$ ::= SET OF SecurityExtensionsSubField
```

```
tySecurityExtensionsSubFieldxe "ty:SecurityExtensionsSubField"$ ::= ExtensionField
```

-- EDIN Initiator

```
tyEDINInitiatorFieldxe "ty:EDINInitiatorField"$ ::= ENUMERATED {  
    internal-ua (0),  
    external-ua (1),  
    internal-ms (2)}
```

-- Notification Extensions

```
tyNotificationExtensionsFieldxe "ty:NotificationExtensionsField"$ ::= SET OF NotificationExtensionsSubField
```

```
tyNotificationExtensionsSubFieldxe "ty:NotificationExtensionsSubField"$ ::= ExtensionField
```

-- Positive Notification fields

```
tyPositiveNotificationFieldsxe "ty:PositiveNotificationFields"$ ::= SEQUENCE {  
    pn-common-fields [0] CommonFields,  
    pn-supplementary-information [1] EDISupplementaryInformation OPTIONAL,  
    pn-extensions [2] PNExtensionsField OPTIONAL }
```

-- Positive Notification Extensions

```
tyPNExtensionsFieldxe "ty:PNExtensionsField"$ ::= SET OF PNExtensionsSubField
```

```
tyPNExtensionsSubFieldxe "ty:PNExtensionsSubField"$ ::= ExtensionField
```

-- Negative notification fields

```
tyNegativeNotificationFieldsxe "ty:NegativeNotificationFields"$ ::= SEQUENCE {  
    nn-common-fields [0] CommonFields,  
    nn-reason-code [1] NNReasonCodeField,  
    nn-supplementary-information [2] EDISupplementaryInformation OPTIONAL,  
    nn-extensions [3] NNExtensionsField OPTIONAL }
```

-- Negative Notification Reason Codes

```
tyNNReasonCodeFieldxe "ty:NNReasonCodeField"$ ::= CHOICE {  
    nn-ua-ms-reason-code [0] NNUAMSReasonCodeField,  
    nn-user-reason-code [1] NNUserReasonCodeField,  
    nn-pdau-reason-code [2] NNPDAUReasonCodeField }
```

-- Negative Notification Reason Codes from an EDI-UA or EDI-MS

```
tyNNUAMSReasonCodeFieldxe "ty:NNUAMSReasonCodeField"$ ::= SEQUENCE {  
    nn-ua-ms-basic-code [0] NNUAMSBasicCodeField,  
    nn-ua-ms-diagnostic [1] NNUAMSDiagnosticField OPTIONAL }
```

- *Negative Notification Basic Reason Codes from an EDI-UA or EDI-MS. These codes are*
- *those specified in Annex B of Recommendation F.435*
- *for the element of service "EDI Notification Request".*

```

tyNNUAMSBasicCodeFieldxe "ty:NNUAMSBasicCodeField" $ ::= INTEGER{
    ecunspecifiedxe "ec:unspecified" $ (0),
    eccannot-deliver-to-userxe "ec:cannot-deliver-to-user" $ (1),
        -- the EDI Interchange can not be passed on to the user
    ecdelivery-timeoutxe "ec:delivery-timeout" $ (2),
        -- the EDI Interchange could not be passed on to the user within
        -- a specified time limit
    ecmessage-discardedxe "ec:message-discarded" $ (3),
        -- the UA/MS discarded the message before handoff to user
    ecsubscription-terminatedxe "ec:subscription-terminated" $ (4),
        -- recipient's subscription terminated after delivery but before
        -- handoff to user
    ecforwarding-errorxe "ec:forwarding-error" $ (5),
        -- EDI Forwarding was attempted, but failed.
    ecsecurity-errorxe "ec:security-error" $ (6)
        -- security error

    -- physical delivery errors indicated by "cannot-deliver-to-user"
} (0..ub-reason-code)

```

- *Negative Notification Diagnostic Codes from an EDI-UA or EDI-MS*

```

tyNNUAMSDiagnosticFieldxe "ty:NNUAMSDiagnosticField" $ ::= INTEGER {
    -- This field may be used to further specify the error signalled in nn-ua-ms-basic-code
    -- Additional information may be indicated in nn-supplementary-information

    -- general diagnostic codes
    ecprotocol-violationxe "ec:protocol-violation" $ (1),
        -- used if the UA detects a protocol error
    ecedim-originator-unknownxe "ec:edim-originator-unknown" $ (2),
    ecedim-recipient-unknownxe "ec:edim-recipient-unknown" $ (3),
    ecedim-recipient-ambiguousxe "ec:edim-recipient-ambiguous" $ (4),
        -- used if the EDIM recipients or originator are not valid
    ecaction-request-not-supportedxe "ec:action-request-not-supported" $ (5),
        -- used when the action requested by the recipient is not performed
    ecedim-expiredxe "ec:edim-expired" $ (6),
        -- used when the expiry date of the received EDIM occurred before the subject EDIM
        -- was successfully passed to the user or forwarded by the EDI-UA
    ecedim-obsoleteuxe "ec:edim-obsolete" $ (7),
        -- used when the EDIM Identifier of the received EDIM was contained in the Obsolete EDIM field
        -- of a previously received EDIM.
    ecduplicate-edimxe "ec:duplicate-edim" $ (8),
        -- used when the same EDIM is received more than once from the same originator
    ecunsupported-extensionxe "ec:unsupported-extension" $ (9),
        -- used if the EDIM contains an extension which is not supported by the UA
    ecincomplete-copy-rejectedxe "ec:incomplete-copy-rejected" $ (10),
        -- used if the EDI-UA does not accept EDIMs with the Incomplete Copy Indication true
    ecedim-too-large-for-applicationxe "ec:edim-too-large-for-application" $ (11),
        -- used if the EDIM cannot be delivered to the user due to length constraints
    -- forwarding error diagnostic codes
    ecforwarded-edim-not-deliveredxe "ec:forwarded-edim-not-delivered" $ (12),
        -- used when an Non-Delivery Report is received for forwarded EDIM
    ecforwarded-edim-delivery-time-outxe "ec:forwarded-edim-delivery-time-out" $ (13),
        -- used when no Delivery Report is received within a given period
    ecforwarding-loop-detectedxe "ec:forwarding-loop-detected" $ (14),
        -- used if the UA receives an EDIM wich contains a previously forwarded EDIM
    ecunable-to-accept-responsibilityxe "ec:unable-to-accept-responsibility" $ (15),
        -- used if the EDI-UA cannot accept or forward responsibility
}

```

```

-- interchange header diagnostic codes
ecinterchange-sender-unknownxe "ec:interchange-sender-unknown"§ (16),
    -- used when the UA does not recognizes the interchange-sender of the EDI interchange
ecinterchange-recipient-unknownxe "ec:interchange-recipient-unknown"§ (17),
    -- used when the UA cannot find a valid interchange recipient in the Recipient Specifier
ecininvalid-heading-fieldxe "ec:invalid-heading-field"§ (18),
ecininvalid-bodypart-typexe "ec:invalid-bodypart-type"§ (19),
ecininvalid-message-typexe "ec:invalid-message-type"§ (20),
ecininvalid-syntax-idxe "ec:invalid-syntax-id"§ (21),

-- security error diagnostic codes
ecmessage-integrity-failurexe "ec:message-integrity-failure"§ (22),
ecforwarded-message-integrity-failurexe "ec:forwarded-message-integrity-failure"§ (23),
ecunsupported-algorithmxe "ec:unsupported-algorithm"§ (24),
ecdecryption-failedxe "ec:decryption-failed"§ (25),
ectoken-errorxe "ec:token-error"§ (26),
ecunable-to-sign-notificationxe "ec:unable-to-sign-notification"§ (27),
ecunable-to-sign-message-receiptxe "ec:unable-to-sign-message-receipt"§ (28),
ecauthentication-failurexe "ec:authentication-failure"§ (29),
ecsecurity-context-failurexe "ec:security-context-failure"§ (30),
ecmessage-sequence-failurexe "ec:message-sequence-failure"§ (31),
ecmessage-security-labelling-failurexe "ec:message-security-labelling-failure"§ (32),
ecrepudiation-failurexe "ec:repudiation-failure"§ (33),
ecproof-of-failurexe "ec:proof-of-failure"§ (34)
} (1..ub-reason-code)

```

-- Negative Notification Reason Codes from a user

```

tyNNUserReasonCodeFieldxe "ty:NNUserReasonCodeField"§ ::= SEQUENCE {
    nn-user-basic-code                [0] NNUserBasicCodeField,
    nn-user-diagnostic                [1] NNUserDiagnosticField OPTIONAL }

```

-- Negative Notification Basic Reason Codes from a user

```

tyNNUserBasicCodeFieldxe "ty:NNUserBasicCodeField"§ ::= INTEGER {
    ecunspecifiedxe "ec:unspecified"§ (0),
    ecsyntax-errorxe "ec:syntax-error"§ (1),
    -- used when the user discovers a syntax error within the EDI interchange
    ecinterchange-sender-unknownxe "ec:interchange-sender-unknown"§ (2),
    ecinterchange-recipient-unknownxe "ec:interchange-recipient-unknown"§ (3),
    -- used when the UA cannot find a valid interchange recipient in the Recipient Specifier
    ecinvalid-heading-fieldxe "ec:invalid-heading-field"§ (4),
    ecinvalid-bodypart-typexe "ec:invalid-bodypart-type"§ (5),
    ecinvalid-message-typexe "ec:invalid-message-type"§ (6),
    ecfunctional-group-not-supportedxe "ec:functional-group-not-supported"§ (7),
    ecsubscription-terminatedxe "ec:subscription-terminated"§ (8),
    -- unknown to EDIMS-User service
    ecno-bilateral-agreementxe "ec:no-bilateral-agreement"§ (9),
    ecuser-defined-reasonxe "ec:user-defined-reason"§ (10)
} (0..ub-reason-code)

```

-- Negative Notification Diagnostic Reason Codes from a user

```

tyNNUserDiagnosticCodeFieldxe "ty:NNUserDiagnosticCodeField"§ ::= INTEGER
-- Contains reason passed by user when the value of nn-user-basic-code is user-defined-reason.
-- Additional information may be indicated in nn-supplementary-information

```


-- Negative Notification Reason Codes from a PDAU

```
tyNNPDAUReasonCodeFieldxe "ty:NNPDAUReasonCodeField"$ ::= SEQUENCE {  
    nn-pdau-basic-code [0] NNPDAUBasicCodeField,  
    nn-pdau-diagnostic [1] NNPDAUDiagnosticField OPTIONAL }
```

-- Negative Notification Basic Reason Codes from a PDAU

```
tyNNPDAUBasicCodeFieldxe "ty:NNPDAUBasicCodeField"$ ::= INTEGER {  
    ecunspecifiedxe "ec:unspecified"$ (0),  
    ecundeliverable-mailxe "ec:undeliverable-mail"$ (1),  
        -- used if the PDAU determines that it cannot perform physical delivery of the EDIM  
    ecphysical-rendition-not-performedxe "ec:physical-rendition-not-performed"$ (2)  
        -- used if the PDAU cannot perform the physical rendition of the EDIM  
    } (0..ub-reason-code)
```

-- Negative Notification Diagnostic Codes from a PDAU

```
tyNNPDAUDiagnosticFieldxe "ty:NNPDAUDiagnosticField"$ ::= INTEGER {  
    -- This field may be used to further specify the error signalled in nn-pdau-basic-code  
    -- Additional information may be indicated in the nn-supplementary-information  
    ecundeliverable-mail-physical-delivery-address-incorrectxe "ec:undeliverable-mail-physical-delivery-  
address-incorrect"$ (1),  
    ecundeliverable-mail-physical-delivery-office-incorrect-or-invalidxe "ec:undeliverable-mail-physical-  
delivery-office-incorrect-or-invalid"$ (2),  
    ecundeliverable-mail-physical-delivery-address-incompletexe "ec:undeliverable-mail-physical-delivery-  
address-incomplete"$ (3),  
    ecundeliverable-mail-recipient-unknownxe "ec:undeliverable-mail-recipient-unknown"$ (4),  
    ecundeliverable-mail-recipient-deceasedxe "ec:undeliverable-mail-recipient-deceased"$ (5),  
    ecundeliverable-mail-recipient-refused-to-acceptxe "ec:undeliverable-mail-recipient-refused-to-accept"$ (6),  
    ecundeliverable-mail-organization-expiredxe "ec:undeliverable-mail-organization-expired"$ (7),  
    ecundeliverable-mail-recipient-did-not-claimxe "ec:undeliverable-mail-recipient-did-not-claim"$ (8),  
    ecundeliverable-mail-recipient-changed-address-permanentlyxe "ec:undeliverable-mail-recipient-changed-  
address-permanently"$ (9),  
    ecundeliverable-mail-recipient-changed-address-temporarilyxe "ec:undeliverable-mail-recipient-changed-  
address-temporarily"$ (10),  
    ecundeliverable-mail-recipient-changed-temporary-addressxe "ec:undeliverable-mail-recipient-changed-  
temporary-address"$ (11),  
    ecundeliverable-mail-new-address-unknownxe "ec:undeliverable-mail-new-address-unknown"$ (12),  
    ecundeliverable-mail-recipient-did-not-want forwardingxe "ec:undeliverable-mail-recipient-did-not-want  
forwarding"$ (13),  
    ecundeliverable-mail-originator-prohibited-forwardingxe "ec:undeliverable-mail-originator-prohibited-  
forwarding"$ (14),  
    ecphysical-rendition-attributes-not-supportedxe "ec:physical-rendition-attributes-not-supported"$ (15)  
    } (1..ub-reason-code)
```

-- Negative Notification Extension Field(s)

```
tyNNExtensionsFieldxe "ty:NNExtensionsField"$ ::= SET OF NNExtensionsSubField
```

```
tyNNExtensionsSubFieldxe "ty:NNExtensionsSubField"$ ::= ExtensionField
```

-- Forwarded Notification Fields

```
tyForwardedNotificationFieldsxe "ty:ForwardedNotificationFields"$ ::= SEQUENCE {  
    fn-common-fields           [0] CommonFields,  
    forwarded-to               [1] ForwardedTo,  
    fn-reason-code             [2] FNReasonCodeField,  
    fn-supplementary-information [3] EDISupplementaryInformation OPTIONAL,  
    fn-extensions              [4] FNExtensionsField OPTIONAL }
```

-- Forwarded To

```
tyForwardedToxe "ty:ForwardedTo"$ ::= ORName
```

-- Forwarded Reason Code

```
tyFNReasonCodeFieldxe "ty:FNReasonCodeField"$ ::= CHOICE {  
    fn-ua-ms-reason-code      [0] FNUAMSReasonCodeField,  
    fn-user-reason-code       [1] FNUserReasonCodeField,  
    fn-pdau-reason-code       [2] FNPDAUReasonCodeFields }
```

-- Forwarding Notification Reason Codes from an EDI-UA or EDI-MS

```
tyFNUAMSReasonCodeFieldxe "ty:FNUAMSReasonCodeField"$ ::= SEQUENCE {  
    fn-ua-ms-basic-code       [0] FNUAMSBasicCodeField,  
    fn-ua-ms-diagnostic       [1] FNUAMSDiagnosticField OPTIONAL,  
    fn-security-check         [2] FNUAMSSecurityCheckField DEFAULT FALSE }
```

-- Forwarding Notification Basic Reason Codes from an EDI-UA or EDI-MS

```
tyFNUAMSBasicCodeFieldxe "ty:FNUAMSBasicCodeField"$ ::= INTEGER {  
    ecunspecifiedxe "ec:unspecified"$ (0),  
    econward-routingxe "ec:onward-routing"$ (1),  
    -- used whenever the UA decides to re-route the subject EDIM for local reasons  
    ecrecipient-unknownxe "ec:recipient-unknown"$ (2),  
    ecoriginator-unknownxe "ec:originator-unknown"$ (3),  
    ecforwarded-by-edi-msxe "ec:forwarded-by-edi-ms"$ (4)  
    } (0..ub-reason-code)
```

-- Forwarding Notification Diagnostic Reason Codes from an EDI-UA or EDI-MS

```
tyFNUAMSDiagnosticFieldxe "ty:FNUAMSDiagnosticField"$ ::= INTEGER {  
    -- This field may be used to further specify the error signalled in fn-ua-ms-basic-code.  
    -- Additional information may be indicated in fn-supplementary-information.  
    ecrecipient-name-changedxe "ec:recipient-name-changed"$ (0),  
    ecrecipient-name-deletedxe "ec:recipient-name-deleted"$ (1)  
    } (1..ub-reason-code)
```

-- Forwarding Notification Security Check Codes from an EDI-UA or EDI-MS

```
tyFNUAMSSecurityCheckFieldxe "ty:FNUAMSSecurityCheckField"$ ::= BOOLEAN
```

-- Forwarding Notification Reason Codes from a user

```
tyFNUserReasonCodeFieldxe "ty:FNUserReasonCodeField"$ ::= SEQUENCE {  
    fn-user-basic-code        [0] FNUserBasicCodeField,  
    fn-user-diagnostic        [1] FNUserDiagnosticField OPTIONAL }
```

-- Forwarding Notification Basic Reason Codes from a user

```
ty FNUserBasicCodeFieldxe "ty:FNUserBasicCodeField"$ ::= INTEGER {  
    ecunspecifiedxe "ec:unspecified"$ (0),  
    ecforwarded-for-archivingxe "ec:forwarded-for-archiving"$ (1),  
    ecforwarded-for-informationxe "ec:forwarded-for-information"$ (2),  
    ecforwarded-for-additional-actionxe "ec:forwarded-for-additional-action"$ (3),  
    ecsubscription-changedxe "ec:subscription-changed"$ (4),  
    echeading-field-not-supportedxe "ec:heading-field-not-supported"$ (5),  
    ecbodypart-type-not-supportedxe "ec:bodypart-type-not-supported"$ (6),  
    ecmessage-type-not-supportedxe "ec:message-type-not-supported"$ (7),  
    ecsyntax-identifier-not-supportedxe "ec:syntax-identifier-not-supported"$ (8),  
    ecinterchange-sender-unknownxe "ec:interchange-sender-unknown"$ (9),  
    ecinterchange-sender-unknownxe "ec:interchange-sender-unknown"$ (10),  
    ecuser-defined-reasonxe "ec:user-defined-reason"$ (11)  
} (0..ub-reason-code)
```

-- Forwarding Notification Diagnostic Reason Codes from a user

```
ty FNUserDiagnosticFieldxe "ty:FNUserDiagnosticField"$ ::= INTEGER  
    -- Contains reason passed by user when value of fn-user-basic-code is user-defined-reason.  
    -- Additional information may be indicated in fn-supplementary-information.
```

-- Forwarding Notification Reason Codes from a PDAU

```
ty FNPDAUReasonCodeFieldxe "ty:FNPDAUReasonCodeField"$ ::= SEQUENCE {  
    fn-pdau-basic-code [0] FNPDAUBasicCodeField,  
    fn-pdau-diagnostic [1] FNPDAUDiagnosticField OPTIONAL }
```

-- Forwarding Notification Basic Reason Codes from a PDAU

```
ty FNPDAUBasicCodeFieldxe "ty:FNPDAUBasicCodeField"$ ::= INTEGER {  
    ecunspecifiedxe "ec:unspecified"$ (0),  
    ecforwarded-for-physical-rendition-and-deliveryxe "ec:forwarded-for-physical-rendition-and-delivery"$  
    (1)  
} (0..ub-reason-code)
```

-- Forwarding Notification Diagnostic Reason Codes from a PDAU

```
ty FNPDAUDiagnosticFieldxe "ty:FNPDAUDiagnosticField"$ ::= INTEGER
```

-- Forwarded Notification Extensions

```
ty FNExtensionsFieldxe "ty:FNExtensionsField"$ ::= SET OF FNExtensionsSubField  
  
ty FNExtensionsSubFieldxe "ty:FNExtensionsSubField"$ ::= ExtensionField  
  
END -- of EDIMSIInformationObjects
```

ANNEX C

Reference definition of Message Store Attributes

(This annex forms an integral part of this Recommendation.)

This annex defines for reference purposes the MS attributes specific to EDIM Messaging. It uses the ATTRIBUTE macro of Recommendation X.501.

```
-----  
moEDIMSMMessageStoreAttributesxe "mo:EDIMSMMessageStoreAttributes"§ {joint-iso-ccitt  
    mhs-motis(6) edims(7) modules(0) message-store-attributes(4) }  
  
DEFINITIONS IMPLICIT TAGS ::=  
BEGIN  
  
-- Prologue  
  
-- Exports everything.  
  
IMPORTS  
  
-- EDIMS Object Identifiers  
  
id-bat-body, id-bat-edi-body-part, id-bat-edim-body-part,  
    id-bat-externally-defined-body-part-types, id-bat-interchange-length, id-bat-message-data,  
    id-bat-message-parameters, id-hat-acknowledgement-request, id-hat-application-reference,  
    id-hat-cross-referencing-information,  
    id-hat-date-and-time-of-preparation, id-hat-edi-application-security-element,  
    id-hat-edi-application-security-extensions, id-hat-edi-bodypart-type,  
    id-hat-edi-message-type, id-hat-edin-receiver, id-hat-expiry-time,  
    id-hat-heading, id-hat-heading-extensions, id-hat-incomplete-copy,  
    id-hat-interchange-sender, id-hat-obsolete-edims, id-hat-originator, id-hat-processing-priority-code,  
    id-hat-recipients, id-hat-related-messages, id-hat-sensitivity,  
    id-hat-service-string-advice, id-hat-syntax-identifier,  
    id-hat-this-edim, id-nat-edin-originator, id-nat-first-recipient, id-nat-fn-extensions,  
    id-nat-fn-reason-code, id-nat-fn-supplementary-info, id-nat-forwarded-to,  
    id-nat-nn-extensions, id-nat-nn-reason-code,  
    id-nat-nn-supplementary-info, id-nat-notification-extensions, id-nat-notification-security-elements,  
    id-nat-notification-time, id-nat-pn-extensions, id-nat-pn-supplementary-info, id-nat-subject-edim,  
    id-rat-action-request-for-this-recipient, id-rat-authorization-information-for-this-recipient,  
    id-rat-communications-agreement-id-for-this-recipient, id-rat-edi-notification-requests-for-this-recipient,  
    id-rat-edim-reception-security-requests-for-this-recipient,  
    id-rat-interchange-control-reference-for-this-recipient, id-rat-interchange-recipient-for-this-recipient,  
    id-rat-recipient-extensions-for-this-recipient, id-rat-this-recipient,  
    id-rat-recipient-reference-for-this-recipient,  
    id-rat-responsibility-passing-allowed-for-this-recipient,  
    id-rat-test-indicator-for-this-recipient, id-sat-edim-synopsis, id-sat-edims-entry-type  
-----  
FROM EDIMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) object-identifiers(0) }  
  
-- MS Abstract Service  
  
SequenceNumber  
-----  
FROM MSAbstractService {joint-iso-ccitt mhs-motis(6) ms(4) modules(0) abstract-service(1) }
```

-- EDIMS Information Objects

```
AcknowledgementRequestField, ActionRequestField, ApplicationReferenceField,
AuthorizationInformationField, Body, BodyPartReference, CommunicationsAgreementIdField,
CrossReferencingInformationSubField,
DateAndTimeOfPreparationField, EDIApplicationSecurityElementsField, EDIBodyPart,
EDIBodyPartType, EDIMessageTypeFieldSubField, EDINInitiatorField,
EDINOriginatorField, EDINotificationRequestsField, EDINReceiverField,
EDISupplementaryInformation, ExpiryTimeField,
FirstRecipientField, FNExtensionsSubField, FNReasonCodeField,
ForwardedTo, Heading, HeadingExtensionsSubField, IncompleteCopyField,
InterchangeControlReferenceField, InterchangeRecipientField,
InterchangeSenderField, MessageData, MessageParameters, NNReasonCodeField,
NNExtensionsSubField, NotificationExtensionsSubField, NotificationTimeField,
ObsoleteEDIMsSubField, OriginatorField, PositiveNotificationFields,
PNExtensionsSubField, ProcessingPriorityCodeField, RecipientExtensionsSubField, RecipientField,
RecipientReferenceField, RecipientsSubField, RelatedMessagesField,
ResponsibilityForwarded, ResponsibilityPassingAllowedField, SecurityElementsField,
ServiceStringAdviceField, SubjectEDIMField,
SyntaxIdentifierField, TestIndicatorField, ThisEDIMField
----
FROM EDIMSInformationObjects {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) information-objects(2) }
```

-- IPMS Information Objects

```
ExternallyDefinedParameters
----
FROM IPMSInformationObjects {joint-iso-ccitt mhs-motis(6) ipms(1) modules(0) information-objects(2) }
```

-- Directory Information Framework

```
ATTRIBUTE
----
FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework(1) };
```

-- END imports

-- MESSAGE STORE ATTRIBUTES

-- Summary Attributes

-- EDIMS Entry Type

```
vaedims-entry-typexe "va:edims-entry-type"$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EDIMSEntryType
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-sat-edims-entry-type

tyEDIMSEntryTypexe "ty:EDIMSEntryType"$ ::= ENUMERATED {
    edim (0),
    pn (1),
    nn (2),
    fn (3) }
```

-- EDIM Synopsis

```
vaedim-synopsisxe "va:edim-synopsis"$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EDIMSynopsis
SINGLE VALUE
::= id-sat-edim-synopsis

tyEDIMSynopsisxe "ty:EDIMSynopsis"$ ::= SEQUENCE OF BodyPartSynopsis
```

```

tyBodyPartSynopsisxe "ty:BodyPartSynopsis"$ ::= CHOICE {
    message                                [0] MessageBodyPartSynopsis,
    non-message                            [1] NonMessageBodyPartSynopsis }

tyMessageBodyPartSynopsisxe "ty:MessageBodyPartSynopsis"$ ::= SEQUENCE {
    number                                [0] SequenceNumber,
    synopsis                              [1] EDIMSynopsis }

tyNonMessageBodyPartSynopsisxe "ty:NonMessageBodyPartSynopsis"$ ::= SEQUENCE {
    type                                  [0] OBJECT IDENTIFIER,
    parameters                            [1] ExternallyDefinedParameters OPTIONAL,
    size [2] INTEGER,
    processed                             [3] BOOLEAN DEFAULT FALSE }

```

-- EDI Notification Indicator

```

vaedi-notification-indicatorxe "va:edi-notification-indicator"$ ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX EDINotificationIndicator DEFAULT (0)
    MATCHES FOR EQUALITY
    MULTI VALUE ::=
    id-sat-edi-notification-indicator

```

```

EDINotificationIndicator ::= ENUMERATED {
    no-notification-sent                (0),
    pn-sent                             (1),
    nn-sent                             (2),
    fn-sent                             (3) }

```

-- Heading Attributes

-- Heading

```

vaheadingxe "va:heading"$ ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX Heading
    SINGLE VALUE
    ::= id-hat-heading

```

-- Heading Fields

```

vathis-edimxe "va:this-edim"$ ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ThisEDIMField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-hat-this-edim

```

```

vaoriginatorxe "va:originator"$ ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX OriginatorField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-hat-originator

```

```

vaedin-receiverxe "va:edin-receiver"$ ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX EDINReceiverField
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-hat-edin-receiver

```

```

varesponsibility-forwardedxe "va:responsibility-forwarded"$ ATTRIBUTE
    WITH ATTRIBUTE-SYNTAX ResponsibilityForwarded
    MATCHES FOR EQUALITY
    SINGLE VALUE
    ::= id-hat-responsibility-forwarded

```

vaedi-bodypart-typexe "va:edi-bodypart-type"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX EDIBodyPartType
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-edi-bodypart-type

vaincomplete-copyxe "va:incomplete-copy"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX IncompleteCopyField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-incomplete-copy

expiry-time; **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX ExpiryTimeField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-expiry-time

varrelated-messagesxe "va:related-messages"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX RelatedMessagesReference
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-related-messages

vaobsoleted-edimsxe "va:obsoleted-edims"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX ObsoletedEDIMsSubfield
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-obsoleted-edims

vaedi-application-security-elementxe "va:edi-application-security-element"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX EDIApplicationSecurityElement
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-edi-application-security-element

vaedi-application-security-extensionsxe "va:edi-application-security-extensions"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX EDIApplicationSecurityExtension
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-edi-application-security-extensions

vacross-referencing-informationxe "va:cross-referencing-information"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX CrossReferencingInformationSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-cross-referencing-information

vaedi-message-typexe "va:edi-message-type"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX EDIMessageTypeFieldSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-edi-message-type

vaservice-string-advicexe "va:service-string-advice"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX ServiceStringAdviceField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-service-string-advice

vasyntax-identifierxe "va:syntax-identifier"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX SyntaxIdentifierField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-syntax-identifier

va**interchange-sender****x****e** "va:interchange-sender"§ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX **InterchangeSenderField**
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-hat-interchange-sender

va**date-and-time-of-preparation****x****e** "va:date-and-time-of-preparation"§ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX **DateAndTimeOfPreparationField**
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-hat-date-and-time-of-preparation

va**application-reference****x****e** "va:application-reference"§ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX **ApplicationReferenceField**
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-hat-application-reference

va**heading-extensions****x****e** "va:heading-extensions"§ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX **HeadingExtensionsSubField**
MATCHES FOR EQUALITY
MULTI VALUE
::= id-hat-heading-extensions

-- Recipient Sub-field

va**this-recipient****x****e** "va:this-recipient"§ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX **RecipientField**
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-this-recipient

va**action-request-for-this-recipient****x****e** "va:action-request-for-this-recipient"§ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX **ActionRequestField**
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-action-request-for-this-recipient

va**edi-notification-requests-for-this-recipient****x****e** "va:edi-notification-requests-for-this-recipient"§ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX **EDINotificationRequests**
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-edi-notification-requests-for-this-recipient

va**edi-notification-security-for-this-recipient****x****e** "va:edi-notification-security-for-this-recipient"§ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX **EDINotificationSecurity**
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-edi-notification-security-for-this-recipient

va**edi-reception-security-for-this-recipient****x****e** "va:edi-reception-security-for-this-recipient"§ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX **EDIReceptionSecurity**
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-edi-reception-security-for-this-recipient

va**responsibility-passing-allowed-for-this-recipient****x****e** "va:responsibility-passing-allowed-for-this-recipient"§ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX **ResponsibilityPassingAllowedField**
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-responsibility-passing-allowed-for-this-recipient

-- Fields from EDIFACT interchange

vainterchange-recipient-for-this-recipientxe "va:interchange-recipient-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX InterchangeRecipientField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-interchange-recipient-for-this-recipient

varecipient-reference-for-this-recipientxe "va:recipient-reference-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX RecipientReferenceField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-recipient-reference-for-this-recipient

vainterchange-control-reference-for-this-recipientxe "va:interchange-control-reference-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX InterchangeControlReferenceField
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-rat-interchange-control-reference-for-this-recipient

vaprocessing-priority-code-for-this-recipientxe "va:processing-priority-code-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX ProcessingPriorityCodeField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-processing-priority-code-for-this-recipient

vaacknowledgement-request-for-this-recipientxe "va:acknowledgement-request-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX AcknowledgementRequestField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-acknowledgement-request-for-this-recipient

vacomunications-agreement-id-for-this-recipientxe "va:communications-agreement-id-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX CommunicationsAgreementIdField
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-rat-communications-agreement-id-for-this-recipient

vatest-indicator-for-this-recipientxe "va:test-indicator-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX TestIndicatorField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-test-indication-for-this-recipient

-- END Fields from EDIFACT

-- Fields from ANSI X12 ISA

vaauthorization-information-for-this-recipientxe "va:authorization-information-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX AuthorizationInformationField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-rat-authorization-information-for-this-recipient

-- END Fields from ANSI X12 ISA

varecipient-extensions-for-this-recipientxe "va:recipient-extensions-for-this-recipient"\$ **ATTRIBUTE**
WITH ATTRIBUTE-SYNTAX RecipientExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-rat-recipient-extensions-for-this-recipient

-- *Body Attributes*

-- *Body*

```
vabodyxe "va:body"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX Body
  SINGLE VALUE
  ::= id-bat-body
```

-- *Body Analyses*

```
vainterchange-lengthxe "va:interchange-length"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX InterchangeLength
  MATCHES FOR ORDERING
  SINGLE VALUE
  ::= id-bat-interchange-length
```

```
tyInterchangeLengthxe "ty:InterchangeLength"$ ::= INTEGER
```

-- *Primary Body Parts*

```
vaedi-body-partxe "va:edi-body-part"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDIBodyPart
  SINGLE VALUE
  ::= id-bat-edi-body-part
```

```
vaedim-body-partxe "va:edim-body-part"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX SequenceNumber -- sequence number of the forwarded EDIM entry.
  SINGLE VALUE
  ::= id-bat-edim-body-part
```

```
vamessage-parametersxe "va:message-parameters"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX MessageParameters
  SINGLE VALUE
  ::= id-bat-message-parameters
```

```
vamessage-dataxe "va:message-data"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX MessageData
  SINGLE VALUE
  ::= id-bat-message-data
```

-- *Externally Defined Body Part Types*

```
vaexternally-defined-body-part-typesxe "va:externally-defined-body-part-types"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX OBJECT IDENTIFIER
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-bat-externally-defined-body-part-types
```

-- *Description of the externally-defined-body-part-types attribute syntax for parameter portion only*

```
tyEDIEexternallyDefinedBodyPartParameterAttributexe "ty:EDIEexternallyDefinedBodyPartParameterAttribute"$ ::=
  SEQUENCE {
    body-part-reference          [0] BodyPartReference OPTIONAL,
    parameter                    [1] ExternallyDefinedParameters }
```

-- Notification Attributes

-- Common Fields

vasubject-edimxe "va:subject-edim" \$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SubjectEDIMField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-subject-edim

vaedin-originatorxe "va:edin-originator" \$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EDINOriginatorField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-edin-originator

vafirst-recipientxe "va:first-recipient" \$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX FirstRecipientField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-first-recipient

vanotification-timexe "va:notification-time" \$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX NotificationTimeField
MATCHES FOR EQUALITY ORDERING
SINGLE VALUE
::= id-nat-notification-time

vanotification-security-elementsxe "va:notification-security-elements" \$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX SecurityElementsField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-notification-security-elements

vaedin-initiatorxe "va:edin-initiator" \$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EDINInitiatorField
MATCHES FOR EQUALITY
SINGLE VALUE
::= id-nat-edin-initiator

vanotification-extensionsex "va:notification-extensions" \$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX NotificationExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-nat-notification-extensions

-- Positive Notification Extension Fields

vapn-supplementary-informationxe "va:pn-supplementary-information" \$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX EDISupplementaryInformation
MATCHES FOR EQUALITY SUBSTRINGS
SINGLE VALUE
::= id-nat-pn-supplementary-info

vapn-extensionsex "va:pn-extensions" \$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX PNExtensionsSubField
MATCHES FOR EQUALITY
MULTI VALUE
::= id-nat-pn-extensions

-- *Negative Notification Fields*

```
vann-reasonxe "va:nn-reason"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NNReasonCodeField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-nn-reason-code

vann-supplementary-informationxe "va:nn-supplementary-information"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDISupplementaryInformation
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-nn-supplementary-info

vann-extensionsex "va:nn-extensions"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX NNExtensionsSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-nat-nn-extensions
```

-- *Forwarded Fields*

```
vaforwarded-toxe "va:forwarded-to"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX ForwardedTo
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-forwarded-to

vafn-reason-codexe "va:fn-reason-code"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX FNReasonCodeField
  MATCHES FOR EQUALITY
  SINGLE VALUE
  ::= id-nat-fn-reason-code

vafn-supplementary-informationxe "va:fn-supplementary-information"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX EDISupplementaryInformation
  MATCHES FOR EQUALITY SUBSTRINGS
  SINGLE VALUE
  ::= id-nat-fn-supplementary-info

vafn-extensionsex "va:fn-extensions"$ ATTRIBUTE
  WITH ATTRIBUTE-SYNTAX FNExtensionsSubField
  MATCHES FOR EQUALITY
  MULTI VALUE
  ::= id-nat-fn-extensions

END -- of EDIMSMMessageStoreAttributes
```

ANNEX D

Reference definition of Message Store Auto-Action Attributes

(This annex forms an integral part of this Recommendation.)

This annex, a supplement to Annex C, defines for reference purposes the MS attributes specific to EDIM Messaging Auto-Actions. It uses the ATTRIBUTE macro of Recommendation X.501.

```
-----  
moEDIMSAutoActionTypesxe "mo:EDIMSAutoActionTypes"$ {joint-iso-ccitt  
    mhs-motis(6) edims(7) modules(0) message-store-auto-actions(7)}  
DEFINITIONS ::=   
BEGIN  
  
-- Prologue  
  
-- Exports everything.  
  
IMPORTS  
  
-- EDIMS Object Identifiers  
  
    id-act-edi-auto-forward  
    ----  
    FROM EDIMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) object-identifiers(0)}  
  
-- EDIMS Information Objects  
  
    EDISupplementaryInformation, RecipientField, ActionRequestField  
    EDINotificationRequestsField, ResponsibilityPassingAllowed  
    ----  
    FROM EDIMSInformationObjects {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) information-objects(2) }  
  
-- MS Abstract Service  
  
    AUTO-ACTION, Filter  
    ----  
    FROM MSAbstractService {joint-iso-ccitt mhs-motis(6) ms(4) modules(0) abstract-service(1)}
```

-- MS General Auto Actions

```
PerMessageAutoForwardFields, PerRecipientAutoForwardFields
----
FROM MSGeneralAutoActionTypes {joint-iso-ccitt mhs-motis(6) ms(4) modules(0) general-auto-action-types(3) }
```

-- MTS Upper Bounds

```
ub-recipients
----
FROM MTSUpperBounds {joint-iso-ccitt mhs-motis(6) mts(3) modules(0) upper-bounds(3) }
```

-- MTS Abstract Service Definition

```
ORName
----
FROM MTSAbstractService {joint-iso-ccitt mhs-motis(6) mts(3) modules(0) mts-abstract-service(1) };
```

-- END Imports

-- Auto-Action Types

-- EDI Auto Forwarding Registration

```
moedi-auto-forwardxe "mo:edi-auto-forward"$ AUTO-ACTION
REGISTRATION PARAMETER IS EDIForwardRegistrationParameter
::= id-act-edi-auto-forward

moEDIForwardRegistrationParameterxe "mo:EDIForwardRegistrationParameter"$ ::= SEQUENCE {
    filter [0] Filter OPTIONAL,
    edi-supplementary-info [1] EDISupplementaryInfo OPTIONAL,
    delete-after-forwarding [2] BOOLEAN DEFAULT FALSE,
    edi-forwarding-mode CHOICE {
        forwarding-with-responsibility-not-accepted [3] ForwardWithRespForwarded,
        forwarding-with-responsibility-accepted [4] ForwardWithRespAccepted }
}
```

Auto Action Registration Parameters for Forwarding with Responsibility not Accepted

```
moForwardWithRespNotAcceptedxe "mo:ForwardWithRespNotAccepted"$ ::= SET {
    COMPONENTS OF PerMessageAutoForwardFields, -- from envelope PerMessageFields
    per-recipient-field [3] PerRecipientAutoForwardFields,
    notification-argument [4] NotificationArguments OPTIONAL }

moNotificationArgumentsxe "mo:NotificationArguments"$ ::= SET {
    COMPONENTS OF PerMessageAutoForwardFields, -- from envelope PerMessageFields
    per-recipients-field [3] SEQUENCE SIZE (1..ub-recipients) OF
        PerRecipientAutoForwardFields }
```

Auto Action Registration Parameters for Forwarding with Responsibility Accepted

```
moForwardWithRespAcceptedxe "mo:ForwardWithRespAccepted"$ ::= SET {
    COMPONENTS OF PerMessageAutoForwardFields, -- from envelope PerMessageFields
    per-recipients-field [3] SEQUENCE SIZE (1..ub-recipients) OF
        PerRecipientAutoForwardFields,
    notification-argument [4] NotificationArguments OPTIONAL,
    heading-fields [5] HeadingFields OPTIONAL }
```

```

moHeadingFieldsxe "mo:HeadingFields"$ ::= SEQUENCE {
    new-edin-receiver-name          [0] ORName OPTIONAL,
    next-recipient                  [1] RecipientField,
    next-recipient-action-request   [2] ActionRequestField DEFAULT {id-for-action},
    next-recipient-edi-notification-requests-field [3] EDINotificationRequestsField OPTIONAL,
    next-responsibility-passing-allowed [4] ResponsibilityPassingAllowedField DEFAULT FALSE }

```

END -- of EDIMSAutoActionTypes

ANNEX E

Reference definition of EDIMS Functional Objects

(This annex forms an integral part of this Recommendation.)

This annex defines for reference purposes the functional objects of EDI Messaging. It uses the OBJECT and REFINE macros of Recommendation X.407.

```

moEDIMSFunctionalObjectsxe "mo:EDIMSFunctionalObjects"$ {joint-iso-ccitt
    mhs-motis(6) edims(7) modules(0) functional-objects(1)}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

```

-- *Prologue*

-- *Exports everything.*

IMPORTS

-- *EDIMS Abstract Service*

```

origination, reception
----
FROM EDIMSAbstractService {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) abstract-service(3)}

```

-- *EDIMS Object Identifiers*

```

id-ot-edime, id-ot-edims, id-ot-edi-ms, id-ot-edi-ua,
id-ot-edimg-user, id-ot-pdau,
id-ref-primary, id-ref-secondary
----
FROM EDIMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) object-identifiers(0)}

```

-- *MS Abstract Service*

```

retrieval
----
FROM MSAbstractService {joint-iso-ccitt mhs-motis(6) ms(4) modules(0) abstract-service(1)}

```

-- *MTS Abstract Service*

```
administration, delivery, mTS, submission
----
FROM MTSAbstractService {joint-iso-ccitt mhs-motis(6) mts(3) modules(0) mts-abstract-service(1)}
```

-- *Abstract service definition conventions*

```
OBJECT, REFINE
----
FROM AbstractServiceNotation {joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1) };
```

-- *END imports*

-- *"Root" Object Type*

```
vaedimeXe "va:edime" § OBJECT ::= id-ot-edime
```

Primary Refinement

```
vaedime-refinementxe "va:edime-refinement" § REFINE edime AS
  edims
    origination [S] PAIRED WITH edimg-user
    reception [S] PAIRED WITH edimg-user
    edimg-user RECURRING
    ::= id-ref-primary
```

-- *Primary Object Types*

-- *EDI User*

```
vaedimg-userxe "va:edimg-user" § OBJECT
  PORTS {
    origination [C],
    reception [C] }
    ::= id-ot-edimg-user
```

EDI Messaging System

```
vaedimsxe "va:edims" § OBJECT
  PORTS {
    origination [S],
    reception [S] }
    ::= id-ot-edims
```


-- Secondary Refinement

```
vaedims-refinementxe "va:edims-refinement"$ REFINE edims AS
  mTS
    submission [S] PAIRED WITH edi-ua, edi-ms
    delivery [S] PAIRED WITH edi-ua, edi-ms
    administration [S] PAIRED WITH edi-ua, edi-ms
  edi-ua RECURRING
    origination [S] VISIBLE
    reception [S] VISIBLE
  edi-ms RECURRING
    submission [S] PAIRED WITH edi-ua
    retrieval [S] PAIRED WITH edi-ua
    administration [S] PAIRED WITH edi-ua
  pdau RECURRING
    reception [S] VISIBLE
  ::= id-ref-secondary
```

-- Secondary Object Types

-- EDI User Agent

```
vaedi-uaxe "va:edi-ua"$ OBJECT
  PORTS {
    origination [S],
    reception [S],
    submission [C],
    delivery [C],
    retrieval [C],
    administration [C] }
  ::= id-ot-edi-ua
```

-- EDI Message Store

```
vaedi-msxe "va:edi-ms"$ OBJECT
  PORTS {
    submission [S],
    retrieval [S],
    administration [S],
    submission [C],
    delivery [C],
    administration [C] }
  ::= id-ot-edi-ms
```

-- Physical Delivery Access Unit

```
vapdauxe "va:pdau"$ OBJECT
  PORTS {
    reception [S] }
  ::= id-ot-pdau
```

END -- of EDIMSFunctionalObjects

ANNEX F

Reference definition of EDIMS Abstract Service

(This annex forms an integral part of this Recommendation.)

This annex defines for reference purposes the EDIMS Abstract Service. It uses the PORT and ABSTRACT-OPERATION and ABSTRACT-ERROR macros of Recommendation X.407.

```
-----
moEDIMSAbstractService "mo:EDIMSAbstractService" § {joint-iso-ccitt
  mhs-motis(6) edims(7) modules(0) abstract-service(3)}
  DEFINITIONS IMPLICIT TAGS ::=
  BEGIN

-- Prologue

-- Exports everything.

  IMPORTS

-- EDIMS Information Objects

  Heading, EDIM, EDIN, InformationObject
  ----
  FROM EDIMSInformationObjects {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) information-objects(2) }

-- EDIMS Object Identifiers

  id-pt-origination, id-pt-reception
  ----
  FROM EDIMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) object-identifiers(0) }

-- MTS Abstract Service

  MessageDeliveryEnvelope, MessageSubmissionEnvelope,
  MessageSubmissionIdentifier, MessageSubmissionTime,
  ProbeSubmissionEnvelope, ProbeSubmissionIdentifier,
  ProbeSubmissionTime, RecipientImproperlySpecified,
  ReportDeliveryEnvelope
  ----
  FROM MTSAbstractService {joint-iso-ccitt mhs-motis(6) mts(3) modules(0) mts-abstract-service(1) }

-- Abstract service definition conventions

  ABSTRACT-ERROR, ABSTRACT-OPERATION, PORT
  ----
  FROM AbstractServiceNotation {joint-iso-ccitt mhs-motis(6) asdc(2) modules(0) notation(1) };
```

-- *Primary Port Types*

-- *Origination*

```
vaoriginationxe "va:origination"$ PORT
CONSUMER INVOKES {
  OriginateProbe,
  OriginateEDIM,
  OriginateEDIN }
::= id-pt-origination
```

-- *Reception*

```
vareceptionxe "va:reception"$ PORT
SUPPLIER INVOKES {
  ReceiveReport,
  ReceiveEDIM,
  ReceiveEDIN }
::= id-pt-reception
```

-- *ABSTRACT OPERATIONS*

-- *Origination Abstract Operations*

-- *Originate Probe*

```
tyOriginateProbexxe "ty:OriginateProbe"$ ::= ABSTRACT-OPERATION
ARGUMENT SET {
  envelope [0] ProbeSubmissionEnvelope,
  content [1] EDIM }
RESULT SET {
  submission-identifier [0] ProbeSubmissionIdentifier,
  submission-time [1] ProbeSubmissionTime }
ERRORS { RecipientImproperlySpecified }
```

-- *Originate EDIM*

```
tyOriginateEDIMxe "ty:OriginateEDIM"$ ::= ABSTRACT-OPERATION
ARGUMENT SET {
  envelope [0] MessageSubmissionEnvelope,
  content [1] EDIM }
RESULT SET {
  submission-identifier [0] MessageSubmissionIdentifier,
  submission-time [1] MessageSubmissionTime }
ERRORS { RecipientImproperlySpecified }
```

-- *Originate EDIN*

```
tyOriginateEDINxe "ty:OriginateEDIN"$ ::= ABSTRACT-OPERATION
ARGUMENT SET {
  envelope [0] MessageSubmissionEnvelope,
  content [1] EDIN }
RESULT SET {
  submission-identifier [0] MessageSubmissionIdentifier,
  submission-time [1] MessageSubmissionTime }
ERRORS { RecipientImproperlySpecified }
```

-- *Reception Abstract Operations*

-- *Receive Report*

```
tyReceiveReportxe "ty:ReceiveReport"$ ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope [0] ReportDeliveryEnvelope,
    undelivered-object [1] InformationObject OPTIONAL }
  RESULT
  ERRORS {}
```

-- *Receive EDIM*

```
tyReceiveEDIMxe "ty:ReceiveEDIM"$ ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope [0] MessageDeliveryEnvelope,
    content [1] EDIM }
  RESULT
  ERRORS {}
```

-- *Receive EDIN*

```
tyReceiveEDINxe "ty:ReceiveEDIN"$ ::= ABSTRACT-OPERATION
  ARGUMENT SET {
    envelope [0] MessageDeliveryEnvelope,
    content [1] EDIN }
  RESULT
  ERRORS {}
```

END -- of EDIMSAbstractService

ANNEX G

Reference definition of EDIMS Upper Bounds Parameters

(This annex forms an integral part of this Recommendation.)

This annex defines for reference purposes the upper bounds of various variable-length information items whose abstract syntaxes are defined in the ASN.1 modules of prior annexes.

```
moEDIMSupperBoundsex "mo:EDIMSupperBounds"$ { joint-iso-ccitt
  mhs-motis(6) edims(7) modules(0) upper-bounds(5) }
  DEFINITIONS ::=
  BEGIN
```

-- Prologue

-- Exports everything.

IMPORTS -- nothing -- ;

-- Upper bounds

```
ubub-application-reference "ub:ub-application-reference" $ INTEGER ::= 14
ubub-authorization-information "ub:ub-authorization-information" $ INTEGER ::= 10
ubub-authorization-information-qualifier "ub:ub-authorization-information-qualifier" $ INTEGER ::= 2
ubub-communications-agreement-id "ub:ub-communications-agreement-id" $ INTEGER ::= 35
ubub-edl-association-assigned-code "ub:ub-edl-association-assigned-code" $ INTEGER ::= 6
ubub-edl-application-security-elements "ub:ub-edl-application-security-elements" $ INTEGER ::= 8191
ubub-edl-controlling-agency "ub:ub-edl-controlling-agency" $ INTEGER ::= 2
ubub-edl-document-release "ub:ub-edl-document-release" $ INTEGER ::= 3
ubub-edl-document-version "ub:ub-edl-document-version" $ INTEGER ::= 3
ubub-edl-message-type "ub:ub-edl-message-type" $ INTEGER ::= 6
ubub-identification-code-qualifier "ub:ub-identification-code-qualifier" $ INTEGER ::= 4
ubub-identification-code "ub:ub-identification-code" $ INTEGER ::= 35
ubub-interchange-control-reference "ub:ub-interchange-control-reference" $ INTEGER ::= 14
ubub-local-reference "ub:ub-local-reference" $ INTEGER ::= 64
ubub-processing-priority-code "ub:ub-processing-priority-code" $ INTEGER ::= 1
ubub-reason-code "ub:ub-reason-code" $ INTEGER ::= 32767
ubub-recipient-reference-qualifier "ub:ub-recipient-reference-qualifier" $ INTEGER ::= 2
ubub-recipient-reference "ub:ub-recipient-reference" $ INTEGER ::= 14
ubub-routing-address "ub:ub-routing-address" $ INTEGER ::= 14
ubub-syntax-identifier "ub:ub-syntax-identifier" $ INTEGER ::= 4
ubub-syntax-version "ub:ub-syntax-version" $ INTEGER ::= 5
END -- of EDIMSupperBounds
```

ANNEX H

Reference definition of Directory Object Classes and Attributes

(This annex forms an integral part of this Recommendation.)

This annex defines for reference purposes the object identifiers, object classes, attributes, and attribute syntaxes specific to EDI use of Directory. It uses the OBJECT-CLASS, ATTRIBUTE, and ATTRIBUTE-SYNTAX macros of Recommendation X.501. Annex J contains a discussion and description of the objects defined here.

```
-----
moEDIUseOfDirectoryxe "mo:EDIUseOfDirectory" ${joint-iso-ccitt
    mhs-motis(6) edims(7) modules(0) edi-directory-cl-att(6) }
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- Prologue

-- Exports everything

IMPORTS

-- EDIMS Object Identifiers

id-dir
----
FROM EDIMSObjectIdentifiers {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) object-identifiers(0) }

-- EDIMS Information Objects

EDIBodypartType, EDIMessageTypeFieldSubField, SyntaxIdentifier, SyntaxVersion
----
FROM EDIMSAbstractService { joint-iso-ccitt mhs-motis(6) edims(7) modules(0) information-objects(2) }

-- EDIMS Upper bounds

ub-edi-association-assigned-code, ub-edi-controlling-agency,
ub-edi-document-release, ub-edi-document-version
----
FROM EDIMSupperBounds {joint-iso-ccitt mhs-motis(6) edims(7) modules(0) upper-bounds(5) }

-- MHS Directory Object Classes and Attributes

mhs-user, mhs-user-agent, mhs-message-store
----
FROM MHSDirectoryObjectAndAttributes {joint-iso-ccitt mhs-motis(6) arch(5) modules(0) directrory(1) }
```

-- Information Framework

ATTRIBUTE, ATTRIBUTE-SYNTAX, OBJECT-CLASS

FROM InformationFramework { joint-iso-ccitt ds(5) modules(1) informationFramework(1) }

-- Selected Object Classes

applicationEntity, top

FROM SelectedObjectClasses { joint-iso-ccitt ds(5) modules(1) selectedObjectClasses(6) }

-- Selected Attribute Types and Syntaxes

caseExactStringSyntax

FROM SelectedAttributeTypes { joint-iso-ccitt ds(5) modules(1) selectedAttributeTypes(5) };

-- END Imports

-- OBJECT IDENTIFIER ASSIGNMENTS FOR USE OF DIRECTORY

-- Categories

idid-docxe "id:id-doc"\$

idid-datxe "id:id-dat"\$

idid-dasxe "id:id-das"\$

ID ::= {id-dir 0} -- directory object classes

ID ::= {id-dir 1} -- directory attribute types

ID ::= {id-dir 2} -- directory attribute syntaxes

-- Directory Object Classes

idid-doc-edi-userxe "id:id-doc-edi-user"\$

ID ::= {id-doc 0}

idid-doc-edi-user-agentxe "id:id-doc-edi-user-agent"\$

ID ::= {id-doc 1}

idid-doc-edi-message-storexe "id:id-doc-edi-message-store"\$

ID ::= {id-doc 2}

-- Directory Attribute Types

idid-dat-edi-namexe "id:id-dat-edi-name"\$

ID ::= {id-dat 0}

idid-dat-edi-routing-addressxe "id:id-dat-edi-routing-address"\$

ID ::= {id-dat 1}

idid-dat-edi-capabilitiesxe "id:id-dat-edi-capabilities"\$

ID ::= {id-dat 2}

-- Directory Attribute Syntaxes

idid-das-edi-capabilitiesxe "id:id-das-edi-capabilities"\$

ID ::= {id-das 0}

-- END Object Identifier Assignments

-- *Object Classes for EDI Use of Directory*

-- *EDI User*

```
vaedi-userxe "va:edi-user"$ OBJECT CLASS
SUBCLASS OF top
MUST CONTAIN {edi-name}
MAY CONTAIN {edi-routing-address, edi-capabilities}
::= id-doc-edi-user
```

-- *EDI User Agent*

```
vaedi-user-agentxe "va:edi-user-agent"$ OBJECT-CLASS
SUBCLASS OF mhs-user-agent
MAY CONTAIN {edi-capabilities}
::= id-doc-edi-user-agent
```

-- *EDI Message Store*

```
vaedi-message-storexe "va:edi-message-store"$ OBJECT-CLASS
SUBCLASS OF mhs-message-store
MAY CONTAIN {edi-capabilities}
::= id-doc-edi-message-store
```

-- *ATTRIBUTES*

-- *EDI Name*

```
vaedi-namexe "va:edi-name"$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX caseExactStringSyntax
SINGLE VALUE
::= id-dat-edi-name

-- The edi-name shall be one of the following:
-- * a name assigned by an EDI naming authority, e.g. the Sender-ID or the Receiver-ID,
-- * a name assigned by the EDI user's organization.
```

-- *EDI Routing Address*

```
vaedi-routing-addressxe "va:edi-routing-address"$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX caseExactStringSyntax
SINGLE VALUE
::= id-dat-edi-routing-address

-- The term edi-routing-address reflects its derivation from a data element in the
-- EDI Interchange with the same name.
```

-- *EDI Capabilities*

```
vaedi-capabilitiesxe "va:edi-capabilities"$ ATTRIBUTE
WITH ATTRIBUTE-SYNTAX edi-capabilities-syntax
MULTI VALUE
::= id-dat-edi-capabilities
```


-- ATTRIBUTE SYNTAXES

-- EDI Capabilities Syntax

```
vaedi-capabilities-syntaxxe "va:edi-capabilities-syntax"$ ATTRIBUTE-SYNTAX
  EDIUserCapability
  MATCHES FOR EQUALITY
  ::= id-das-edi-capabilities

tyEDIUserCapabilityxe "ty:EDIUserCapability"$ ::= SEQUENCE {
  edi-bodypart-type          [0] EDIBodypartType OPTIONAL
  edi-processable-document   [1] EDIProcessableDocument OPTIONAL }

tyEDIProcessableDocumentxe "ty:EDIProcessableDocument"$ ::= SEQUENCE {
  standardVersion            [0] SyntaxVersion OPTIONAL,
  standardSyntaxId           [1] SyntaxIdentifier OPTIONAL,
  documentType               [2] EDIMessageTypeFieldSubField OPTIONAL,
  documentVersion            [3] DocumentVersion OPTIONAL,
  documentRelease            [4] DocumentRelease OPTIONAL,
  controllingAgency         [5] ControllingAgency OPTIONAL,
  associationAssignedCode     [6] AssociationAssignedCode OPTIONAL }

vaAssociationAssignedCodexe "va:AssociationAssignedCode"$ ::= T61String (SIZE(1..ub-edi-association-assigned-code))

vaControllingAgencyxe "va:ControllingAgency"$ ::= T61String (SIZE(1..ub-edi-controlling-agency))

vaDocumentReleasexe "va:DocumentRelease"$ ::= NumericString (SIZE(1..ub-edi-document-release))

vaDocumentVersionxe "va:DocumentVersion"$ ::= NumericString (SIZE(1..ub-edi-document-version))

END -- EDIMUseOfDirectory module
```

ANNEX I

Enhanced Security Model

(This annex forms an integral part of this Recommendation.)

1.1 Introduction

This annex describes the enhancements required to the security model defined in Recommendation X.402.

1.2 Security Services

The additional security services and pervasive mechanisms described in Recommendation F.435 require the security model defined in 10 of Recommendation X.402 to be enhanced with the following security services:

- Non-repudiation/Proof of Reception;
- Non-repudiation/Proof of Retrieval;
- Non-repudiation/Proof of Transfer;
- Non-repudiation of Content.

1.3 Enhancements to Clause 10.2: Security Services

1.3.1 Changes to Recommendation X.402

Changes to Table 7 of Recommendation X.402 are shown in Table I.1. Two new classes of services are added; these are EDIM Responsibility Authentication and Non-repudiation of EDIM Responsibility.

1.3.2 EDIM Responsibility authentication services

1.3.2.1 Proof of EDI Notification

This security service enables the originator of a message to obtain corroboration that his message has been received, and EDIM Responsibility has been accepted, forwarded, or refused.

This service may be provided by using the Content Integrity check on message submission applied to the EDI Notification of the subject EDIM.

1.3.2.2 Proof of retrieval

This security service enables the MS administrator to obtain corroboration that a particular message has been retrieved from the EDI-MS by the EDI-UA.

Implementation of this security service is a local issue. Additional pervasive mechanisms described in Recommendation F.435 may be used to provide this service.

1.3.2.3 Proof of transfer

This security service enables an MTA or an MD to obtain corroboration that a message has been transferred (relayed) to another MTA or MD. Implementation of this security service is a local issue. Additional pervasive mechanisms described in Recommendation F.435 may be used to provide this service.

1.4 Non-repudiation of EDIM Responsibility services

1.4.1 Non-repudiation of EDI Notification

This security service provides the Originator of a message with irrevocable proof that the message has been received, and EDIM Responsibility has been accepted, forwarded, or refused.

1.4.2 Non-repudiation of retrieval

This security service provides the EDI-MS administrator and the EDI-UA with irrevocable proof that a message has been retrieved from the EDI-MS by the EDI-UA. Implementation of this security service is a local issue. Additional pervasive mechanisms described in Recommendation F.435 may be used to provide this service.

1.4.3 Non-repudiation of transfer

This security service provides an MTA or an MD with irrevocable proof that a message has been transferred (relayed) to another MTA or MD. Implementation of this security service is a local issue. Additional pervasive mechanisms described in Recommendation F.435 may be used to provide this service.

1.4.4 Non-repudiation of Content

This security service provides an EDIMG user with irrevocable proof of the authenticity and integrity of the content of the message.

This security service may be provided in two ways: (1) using a Notarisation Mechanism, or (2) using the Non-repudiation of Origin security service applied to the subject message and the EDI Notification of the subject message, provided the EDI Notification includes irrevocable proof of the content of the subject message.