

## DICOM Correction Item

Correction Number CP-494	
Log Summary: Clarification to Asynchronous Operations Window Negotiation	
Type of Modification Clarification	Name of Standard PS 3.7 2004
Rationale for Correction: <p>The explanation of the use of Asynchronous Operations Window is not clear in PS 3.7. Various interpretations could be made that would affect interoperability. Also, there is no statement concerning the ordering of response messages when Asynchronous Operations are negotiated.</p> <p>This change proposal is written based on the following assumptions. For an SCU proposing an association, the maximum number of operations invoked would mean the maximum number of request messages sent to the server before processing a response message. The maximum number of operations performed for an SCU would be the maximum number of notifications (eg, N-EVENT-REPORT-RQ messages) that could be received before the SCU would send a response message. For an SCP proposing an association, it is assumed the maximum number of operations invoked would be the maximum number of notifications sent, and the maximum number of operations performed would be the maximum number of request messages processed by the SCP.</p> <p>Furthermore, it is assumed that the response messages do not have to be sent in the order that the request messages were received. The use of Priority (0000,0700) in the C-STORE-RQ message fields implies some requests may be processed and a response sent before other requests if they have a higher priority setting.</p>	
Sections of documents affected PS 3.7, Section 7.3, D.3.3.3	
Correction Wording:	

*PS 3.7, Section 7.3, Clarify the possibility of out of order responses*

### 7.3 Service modes

Operations and notifications, on an Association, are used in one of the following two modes:

- a)synchronous
- b)asynchronous

In the synchronous mode, the invoking DIMSE-service-user, on an established Association, requires a response from the performing DIMSE-service-user before invoking another operation or notification.

In the asynchronous mode, the invoking DIMSE-service-user, on an established Association, may continue to invoke further operations or notifications to the performing DIMSE-service-user without awaiting a response. **In the asynchronous mode, the performing DIMSE-service-user may respond to the operations or notifications in a different order than they were received.**

The mode selection (synchronous or asynchronous) is determined at Association establishment time. The synchronous mode serves as the default mode and shall be supported by all DIMSE-service-users. The asynchronous mode is optional and the maximum number of outstanding operations/notifications is negotiated during Association establishment. This negotiation is accomplished by Application Association Information as defined in Annex D.

*Section D.3.3.3, Add notifications to the descriptions and clarify roles and how they relate*

The Asynchronous Operations Window is used to negotiate the maximum number of outstanding operation or sub-operation requests (i.e. command requests) for each direction. The synchronous operations mode is the default mode and shall be support by all DICOM AEs. This negotiation is optional.

The Association-requester conveys in the A-ASSOCIATE request:

- **when negotiating the SCU role for operations,** the maximum number of outstanding operations **it may invoke asynchronously; when negotiating the SCP role for operations, and/or the maximum number of outstanding** sub-operations it may invoke asynchronously; **when negotiating the SCP role for notifications, the maximum number of notifications it may invoke asynchronously**
- **when negotiating the SCP role for operations,** the maximum number of outstanding operations **it may invoke asynchronously; when negotiating the SCU role for operations, and/or the maximum number of outstanding** sub-operations it may perform asynchronously; **when negotiating the SCU role for notifications, the maximum number of notifications it may perform asynchronously when negotiating the SCP role**

A value of zero indicates that the above parameters are unlimited. If the Asynchronous Operations Window is absent the default for the above parameters shall be equal to one.

The Association-acceptor conveys in the A-ASSOCIATE response:

- **when negotiating the SCP role for operations,** the maximum number of outstanding operations; **when negotiating the SCU role for operations, the maximum number of and/or** sub-operations it allows the Association-requester to invoke asynchronously; **when negotiating the SCU role for notifications, the maximum number of outstanding notifications it allows the Association-requester to invoke asynchronously when negotiating the SCU role.** This number shall be equal or less than the number of outstanding **notifications,** operations and/or sub-operations the Association-requester offers to invoke (by the A-ASSOCIATE indication).
- **when negotiating the SCU role for operations,** the maximum number of outstanding operations; **when negotiating the SCP role for operations, the maximum number of and/or** sub-operations it allows the Association-requester to perform asynchronously; **when negotiating the SCP role for notifications, the maximum number of outstanding notifications it allows the Association-requester to perform asynchronously.** This number shall be equal or less than the number of outstanding **notifications,** operations and/or sub-operations the Association-requester offers to perform (by the A-ASSOCIATE indication).

A value of zero indicates that the above parameters are unlimited. If the Asynchronous Operations Window is absent the default for the above parameters shall be equal to one. Figures D.3-5 and D.3-6 illustrate examples of Asynchronous Operations Window negotiation.

If this negotiation is not present in the A-ASSOCIATE indication it shall be omitted in the A-ASSOCIATE response.

Note: The case where the Association-requester offers the value of zero (which indicates unlimited operations), the Association-acceptor may return zero (agreeing to unlimited operations) or negotiate the parameter a