
Article

[Sarah Matthews](#) · Apr 8, 2022 3m read

How To: Implement DIMSE Timeout for a DICOM Production

Introduction

Hi Community,

This article is aimed at developers implementing DICOM productions, specifically for cases with third-party endpoints that cannot handle the DIMSE timeout themselves.

For DICOM applications, the DIMSE timeout is a maximum time to wait for the next DICOM request (-RQ) or a response to a request (-RSP), after an association has been established. Unlike the ARTIM (association establishment) and TXTIM (data transfer) timeouts, the DIMSE timeout lives at a higher level than individual PDUs, with the application-level request/response logic.

The DIMSE timeout is described here in the standard for a few different contexts:

<https://dicom.nema.org/medical/dicom/current/output/chtml/part02/sectE....>

If your production's endpoints can set a DIMSE timeout, then you're all set! However, if not, you can implement a timeout in your production's Process class using the Ens.Alarm mechanism and a counter.

Source:

Example code available in the [community irishealth-ensdemo repo](#).

See the Storage production in the ENSDEMO namespace (Demo.DICOM.Production.Storage) and the corresponding process class.

Details:

Demo.DICOM.Process.Storage.cls, which extends the DICOM Process superclass, controls the logic for the DIMSE timeout. Note the following components:

- A field called RequestTimeout, which specifies the length of the DIMSE timeout in seconds
- A wakeup counter, which tracks the number of pending wakeup calls
- A wakeup call scheduled after a DICOM request is seen and sent to the Operation, in OnMessage:

```
#; Set a wakeup call for this request.  
Set ..WakeupCounter = ..WakeupCounter + 1  
Set tSC = ..ScheduleWakeupCall(..RequestTimeout)  
If $$$ISERR(tSC) $$$LOGSTATUS(tSC)
```

- Similarly, a wakeup call scheduled after an association is established, in OnAssociationEstablished. This is in case the association is created successfully, but the process never sees a request afterward.
- A block at the beginning of OnMessage for handling Ens.AlarmResponse documents:

```
#; DIMSE timeout implemented with the Ens.Alarm request/response mechanism and
```

```

a counter.
    If (pInput.%Extends("Ens.AlarmResponse")) {
        #; Decrement our wakeup counter.
        Set ..WakeupCounter = ..WakeupCounter - 1
        If (..WakeupCounter = 0) {
            #; It's been RequestTimeout seconds since the last request.

            #; Form an abort message
            Set tCommandAbort=##class(EnsLib.DICOM.Command.Abort).%New($$$$ABORTSOURCESERVICEPROVIDER,$$$$ABORTREASONNOTSPECIFIED)

            #; Ensure we try to abort both the operation and service associations
            Set tSC = ..AbortAssociation(..OperationDuplexName,tCommandAbort)
            If $$$ISERR(tSC) $$$LOGSTATUS(tSC)
            Set tSC = ..AbortAssociation(..ServiceDuplexName,tCommandAbort)
            If $$$ISERR(tSC) $$$LOGSTATUS(tSC)

            #; And quit here.
            Quit
        }
        #; At least one wakeup call was scheduled more recently, so this isn't DIMSE timeout.
        Quit
    }
}

```

This business process schedules a new wakeup call when the association is first established, and then every time a new DICOM request is seen and passed to the Operation. ScheduleWakeupCall() sends an Ens.AlarmRequest, which schedules an Ens.AlarmResponse for RequestTimeout seconds later. With each ScheduleWakeupCall(), this process also increments the counter to track how many wakeups are pending. For each wakeup it sees (Ens.AlarmResponse), it decrements the counter. If the wakeup is not the last one, the process discards the wakeup. However, if it is the last wakeup, then the process has encountered the DIMSE timeout, and it aborts the associations on both the Operation and Service sides, as opposed to holding them open and listening indefinitely. If the Store request completes and the Association closes before an AlarmResponse comes back, then late AlarmResponses are discarded automatically.

None of the above is necessary if the endpoints for Storage set the DIMSE timeout themselves and behave correctly, but for extenuating circumstances, this is an example of how to support it from IRIS.

[#Business Process \(BPL\)](#) [#Interoperability](#) [#Ensemble](#) [#InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/how-implement-dimse-timeout-dicom-production>