

DICOM Troubleshooting & Related Issues

Communication Issues

I cannot get these two devices to make and keep a connection long enough to exchange DICOM messages.

Generic Networking Issues

1. I have the wrong address for the device I am trying to communicate with: hostname & port.
2. Another application is already listening on the DICOM port (104, 11112) or on the port I want to use.
3. A firewall somewhere between the client and server systems is blocking communication either because of the TCP port used, or due to some other configuration problem.
4. A slow or unreliable network link is causing mysterious communications failures: DICOM cannot restart or resume an aborted transfer attempt (see 7 below).
5. Router Intrusion Detection Systems classify DICOM traffic as malicious and sever active connections without warning; typically observed as an apparent network failure after dozens of DICOM images have successfully transferred.

DICOM or Application Specific Networking Issues

1. I forgot to find and read the DICOM conformance statement for my device.
I read the conformance statement for my device and it doesn't tell me anything useful.
2. Cannot communicate (query or store) with PACS. The PACS has not been told about the new device.
 - a. Source or destination PACS does not have the new device configured in its table of known application entities, i.e., list of known devices with AE-Title, hostnames, port, etc.
 - b. Source or destination PACS does not have the new device configured with appropriate permissions—is it allowed to store, query, etc.?
3. One or both of the devices do not have useful DICOM log files to describe what data was communicated or what events took place. I need an application in the middle to help follow what is happening.
4. A server does not effectively communicate information about an error back to the client.
 - a. Servers that map every error condition to a single DIMSE status code, e.g., PROCESSING-FAILURE.
 - b. Servers that handle every error condition by issuing an A-Abort PDU and drop the connection.
 - c. Servers and Clients that do not make use of available methods of describing error conditions with allowable return values like error-comment, offending-element, etc.
5. Clients and Servers are not able to send messages or messages are sent in undesired encodings due to transfer syntax support and negotiation issues.
 - a. The sending or receiving device does not support the preferred transfer syntax for sending a particular type of image.
 - b. The sending device does not request the preferred transfer syntax for sending a particular type of image.
 - c. The sending or receiving device claims to support a particular transfer syntax but does not follow the standard correctly.
6. One party to the communication does not properly close an association after an otherwise successful transfer of messages—leaves it open forever or just aborts and goes away silently.
7. One party to the communication fails due to a timeout condition, this may happen for example while sending a large image over a slow network. Perhaps timeouts on the other party need to be adjusted to ensure that the right thing happens.

Reading and Writing Issues

I cannot read or write this data to or from the file system (or network).

1. Can't read or write a DICOM file. Common examples:
 - a. Incompatible or unsupported compression transfer syntax – I need a conversion utility.
 - b. Missing or empty required tags – I need to modify the dataset.
 - c. Invalid data in tags, e.g., decimal string stored as binary data – I need to correct the errant elements.
2. Malformed datasets. Common examples where I need to be able to re-write the dataset:
 - a. Improper padding of odd length tags causes read errors.
 - b. Malformed sequences cannot be read, e.g., CP-246 problem—undefined length sequences must always be encoded as Implicit VR Little-Endian, without regard to the transfer syntax of the containing data set.
 - c. Encapsulated pixel data is stored as a defined length element when it should be stored as undefined length.
3. CD Import – I avoided the auto run feature, but where are the files I need to import? Can I find the “DICOMDIR” file or is it missing? How can I recognize, visualize, or determine what images are on the media?

DICOM Troubleshooting & Related Issues

The data I received is inconsistent with what I have in my database or with what I know should be present

1. Data collisions or inconsistencies in the real-world model, e.g., one study contains series with different accession numbers, Patient IDs, Patient Names, etc.
2. A study was directly copied from a PACS and contains stale header data. The PACS system does not synchronize changes to the stored metadata with the saved DICOM files.
3. I cannot query every item I expected. Getting all the relevant query results out of a PACS when there are more items than the configured “Max Returned Results” set on the PACS can be quite difficult. I might need to query by hour vs. by day.

The data I’ve sent is not ending up in the database in what I would consider a logical way

1. What are the PACS policies for duplicate handling? First one wins? Last one wins? Both are stored? All are quarantined?
2. How are instances grouped into studies? Is it based on time received? Are there timeout settings I need to set? How can I coalesce two partial pieces of a study into one study?
3. Unsupported objects. Some objects are missing from a study that otherwise appeared to store successfully, typically these are objects the PACS does not support, perhaps GSPS or SR objects, e.g., a study with MR and GSPS objects is successfully sent to PACS, but the GSPS objects are silently dropped after arrival. Alternately, the presence of unsupported objects causes the entire study transfer to fail.

I’m sending query requests and the responses I’m getting are not helping me know what’s in the database

1. The PACS won’t tell me certain useful fields, e.g. number-of-study-related-instances.
2. Improper handling of the syntax for Date range queries, especially for single days, e.g., '20130312-', '-20050101', '20081201-20081201' are each valid and should return correct results.
3. Presence of a single non-returnable tag in the query incorrectly fails the entire query to the PACS.

These images do not look right when I load them in my viewer

1. Color images displayed with green tint: YBR-Full image was converted to GBR instead of to RGB.
2. A viewer application is not correctly modifying an image based on data from an associated GSPS or SR object. Is the GSPS or SR object even present in the study? Does the PACS support these presentation state components?

Implementation problems that are unusual enough to be in a category of their own

1. Systems say they support a particular transfer syntax, and they negotiate that syntax at setup time, but they really only support sending in that syntax if the instances they are sending were originally received in that syntax. In other words, they do not have the required CODEC support, and therefore no capability for changing the transfer syntax of the instance. Example: two systems can only agree to talk in ILE (or ELE). So they agree on ILE (or ELE), and then the sender cannot convert its data to the agreed ILE (or ELE) because it was originally received and stored verbatim in another transfer syntax (e.g., JPEG 2000), and now the sender doesn’t know how to convert the original data to ILE (or ELE).
2. Two systems say they support a particular transfer syntax, but they do so in different ways. For example, the group 28 fields for things like Photometric Interpretation, bits allocated/stored, etc. are different than what is contained in or implied by the encapsulated JPEG stream.
3. PACS systems return “Success” immediately as the result of receiving a C-MOVE request, but then secretly, behind the scenes, queue up a request to actually do the move some time later. So they reported that the C-MOVE succeeded when they actually have not even started moving it.
4. A system uses proprietary transfer syntaxes to store DICOM files on disc, making them unreadable by any standard DICOM implementation.