United Lincolnshire Hospitals NHS
NHS Trust

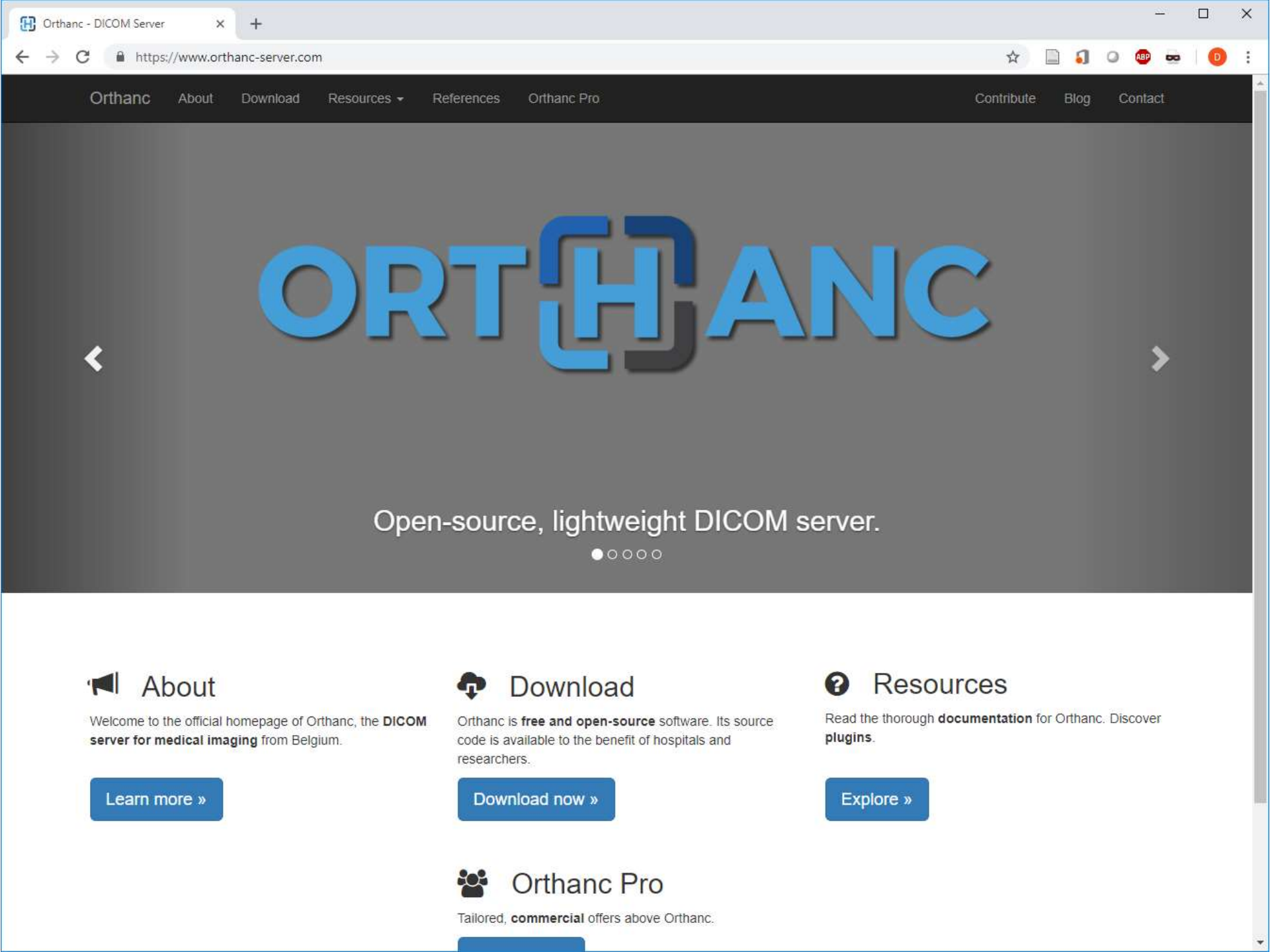
Automatic retrieval of QA images from PACS

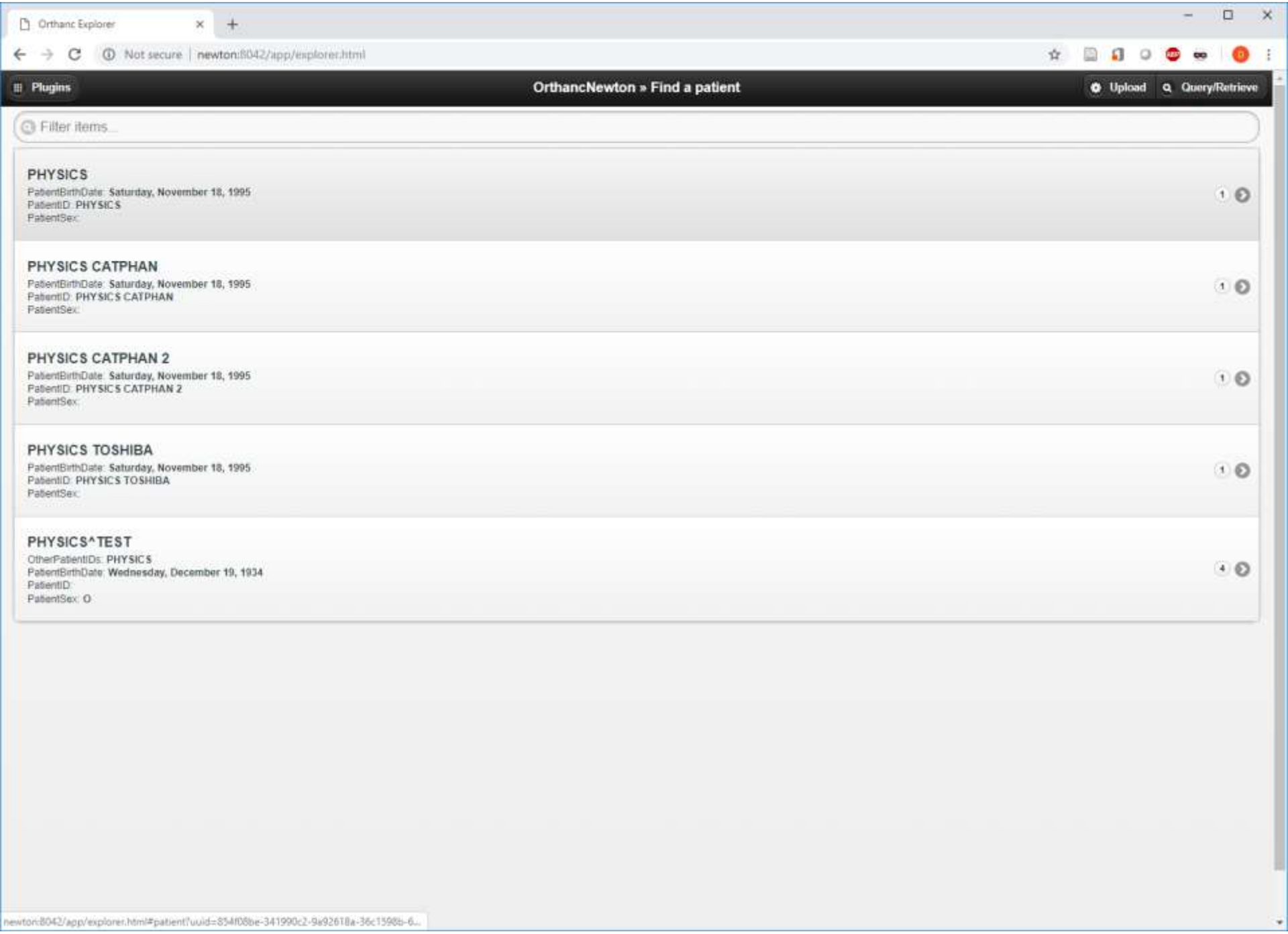
David Platten

Clinical Scientist, Radiation Protection and Radiology Physics

What's been done

- Obtaining images after QA testing can be cumbersome
  - Burn images to CD / DVD with x-ray system computer
  - Send to PACS
    - Burn images to CD / DVD on PACS workstation
    - Export images to network using PACS workstation
- OpenREM's DICOM node is already being used to query PACS
  - Scheduled queries each 24h period for radiography, CT, mammo
  - Added code to check if images are QA-related and forward these to a new QA DICOM store and save them to a network location
  - QA images are then available on the network and in the QA DICOM store the morning after carrying out QA testing
  - Images in the DICOM store can be accessed directly with ImageJ
  - DICOM store web page can be used to query PACS


Orthanc - DICOM Server
https://www.orthanc-server.com
Orthanc
About
Download
Resources
References
Orthanc Pro
Contribute
Blog
Contact
ORTHANC
Open-source, lightweight DICOM server.
About
Welcome to the official homepage of Orthanc, the DICOM server for medical imaging from Belgium.
Learn more »
Download
Orthanc is free and open-source software. Its source code is available to the benefit of hospitals and researchers.
Download now »
Resources
Read the thorough documentation for Orthanc. Discover plugins.
Explore »
Orthanc Pro
Tailored, commercial offers above Orthanc.


Orthanc Explorer
Not secure | newton:8042/app/explorer.html
Plugins
OrthancNewton » Find a patient
Upload
Query/Retrieve
Filter items...
PHYSICS
PatientBirthDate: Saturday, November 18, 1995
PatientID: PHYSICS
PatientSex:
PHYSICS CATPHAN
PatientBirthDate: Saturday, November 18, 1995
PatientID: PHYSICS CATPHAN
PatientSex:
PHYSICS CATPHAN 2
PatientBirthDate: Saturday, November 18, 1995
PatientID: PHYSICS CATPHAN 2
PatientSex:
PHYSICS TOSHIBA
PatientBirthDate: Saturday, November 18, 1995
PatientID: PHYSICS TOSHIBA
PatientSex:
PHYSICS^TEST
OtherPatientIDs: PHYSICS
PatientBirthDate: Wednesday, December 19, 1934
PatientID:
PatientSex: O
newton:8042/app/explorer.html#patient?uuid=854f08be-341990c2-9a92618a-36c1598b-6...

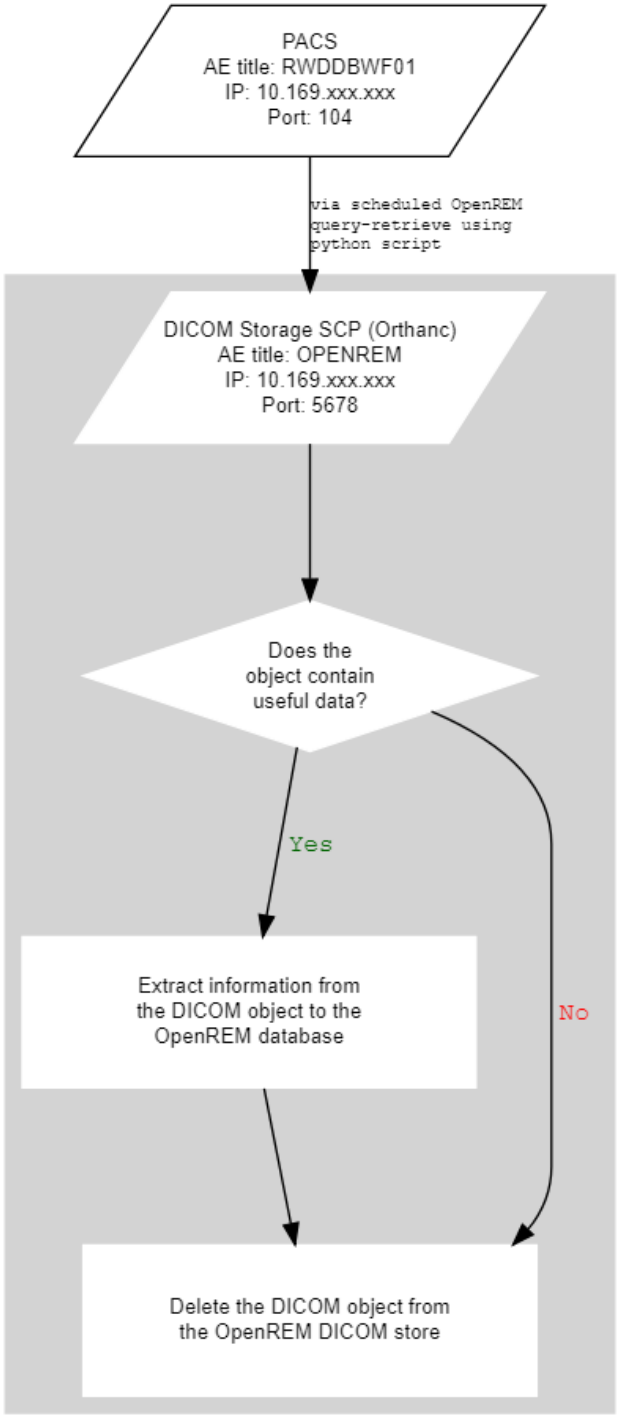
Default OpenREM setup

Query PACS

Retrieve DICOM data

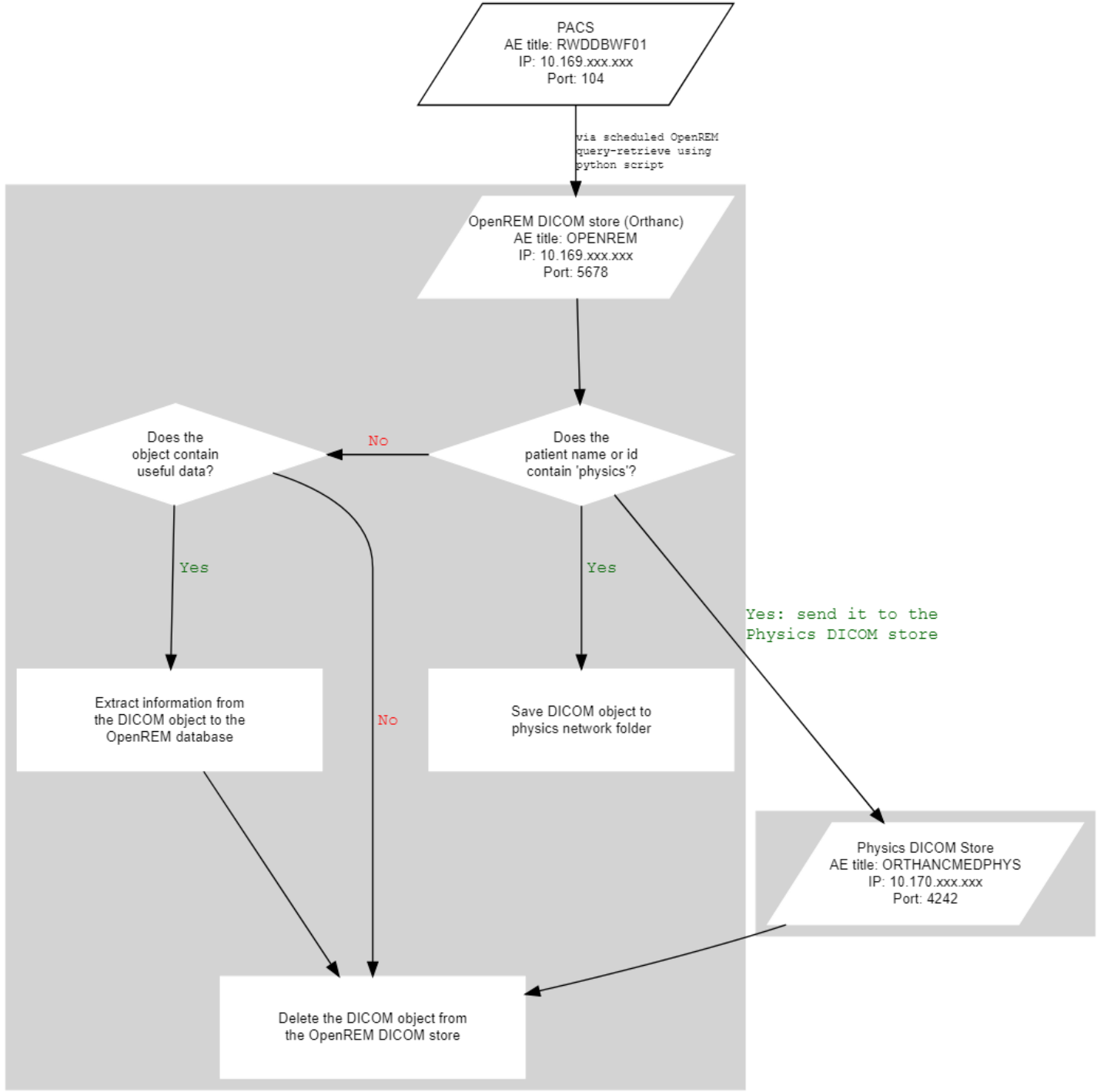
Extract useful information

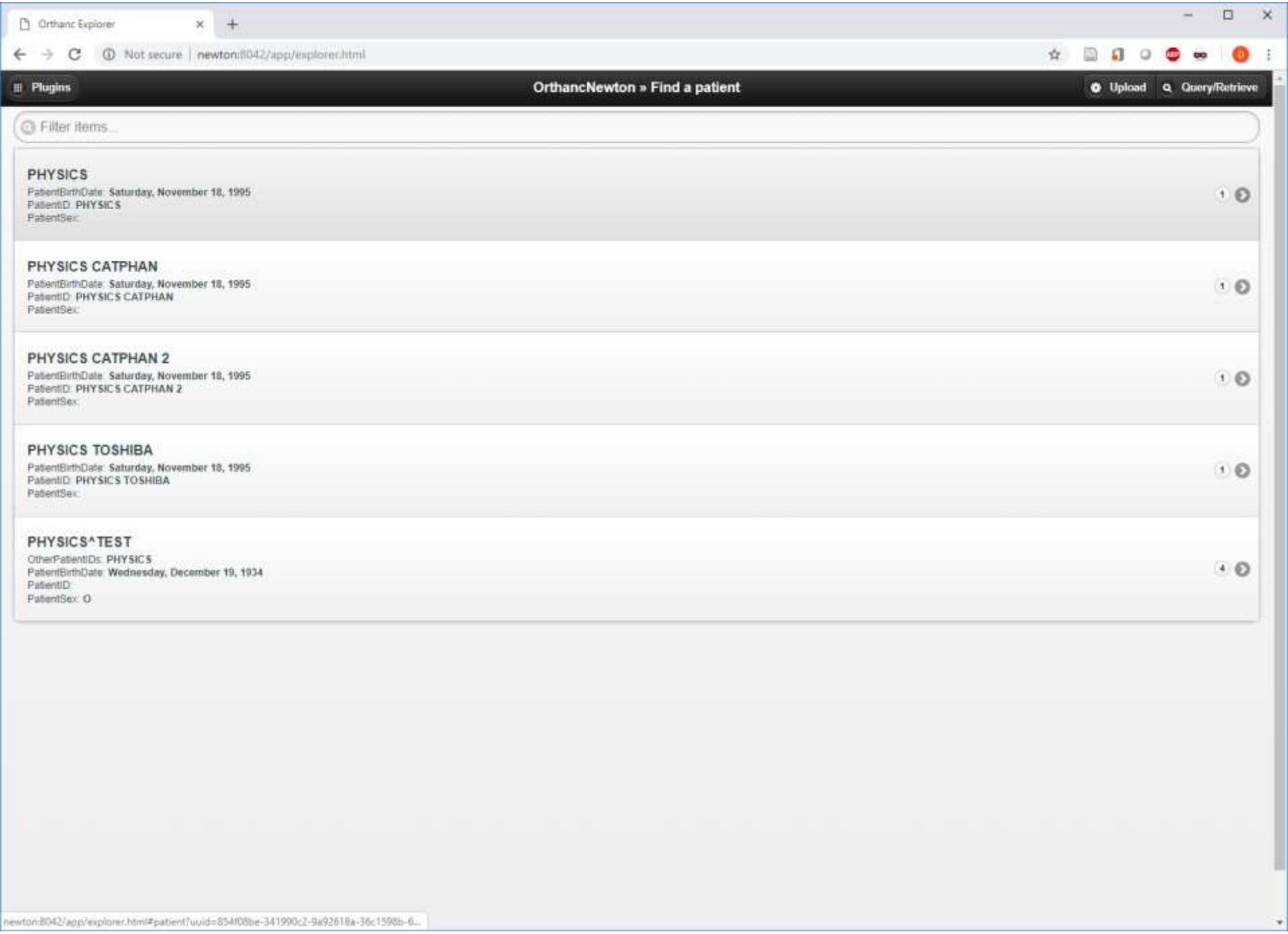
Delete DICOM data


PACS
AE title: RWDDBWF01
IP: 10.169.xxx.xxx
Port: 104
via scheduled OpenREM query-retrieve using python script
DICOM Storage SCP (Orthanc)
AE title: OPENREM
IP: 10.169.xxx.xxx
Port: 5678
Does the object contain useful data?
Yes
No
Extract information from the DICOM object to the OpenREM database
Delete the DICOM object from the OpenREM DICOM store

Modified OpenREM setup

- Query PACS
- Retrieve DICOM data
  - Check if it's "Physics"
- Extract useful information
- Delete DICOM data


PACS
AE title: RWDDBWF01
IP: 10.169.xxx.xxx
Port: 104
via scheduled OpenREM query-retrieve using python script
OpenREM DICOM store (Orthanc)
AE title: OPENREM
IP: 10.169.xxx.xxx
Port: 5678
Does the patient name or id contain 'physics'?
No
Does the object contain useful data?
Yes
Yes
Yes: send it to the Physics DICOM store
Extract information from the DICOM object to the OpenREM database
Save DICOM object to physics network folder
No
Physics DICOM Store
AE title: ORTHANCMEDPHYS
IP: 10.170.xxx.xxx
Port: 4242
Delete the DICOM object from the OpenREM DICOM store


Orthanc Explorer
Not secure | newton:8042/app/explorer.html
Plugins
OrthancNewton » Find a patient
Upload
Query/Retrieve
Filter items...
PHYSICS
PatientBirthDate: Saturday, November 18, 1995
PatientID: PHYSICS
PatientSex:
PHYSICS CATPHAN
PatientBirthDate: Saturday, November 18, 1995
PatientID: PHYSICS CATPHAN
PatientSex:
PHYSICS CATPHAN 2
PatientBirthDate: Saturday, November 18, 1995
PatientID: PHYSICS CATPHAN 2
PatientSex:
PHYSICS TOSHIBA
PatientBirthDate: Saturday, November 18, 1995
PatientID: PHYSICS TOSHIBA
PatientSex:
PHYSICS^TEST
OtherPatientIDs: PHYSICS
PatientBirthDate: Wednesday, December 19, 1934
PatientID:
PatientSex: O
newton:8042/app/explorer.html#patient?uuid=854f08be-341990c2-9a92618a-36c1598b-6...

Modified Orthanc Lua script

Switch on physics filtering
Define what “physics” is
Send “physics” objects to the new node
Save “physics” studies to a folder

This code is on the OpenREM website

```
-- OpenREM python environment and other settings

-- Set this to the path and name of the python executable used by OpenREM
local python_executable = 'D:\\Server_Apps\\python27\\python.exe'

-- Set this to the path of the python scripts folder used by OpenREM
local python_scripts_path = 'D:\\Server_Apps\\python27\\Scripts\\'

-- Set this to the path where you want Orthanc to temporarily store DICOM files
local temp_path = 'E:\\conquest\\dicom\\'

-- Set this to 'mkdir' on Windows, or 'mkdir -p' on Linux
local mkdir_cmd = 'mkdir'

-- Set this to '\\'' on Windows, or '/' on Linux
local dir_sep = '\\'

-- Set this to true if you want Orthanc to keep physics test studies, and have it
-- put them in the physics_to_keep_folder. Set it to false to disable this feature
local use_physics_filtering = true

-- Set this to the path where you want to keep physics-related DICOM images
local physics_to_keep_folder = 'E:\\dicom\\physics\\'

-- Set this to the path and name of your zip utility (used with physics-related images)
local zip_executable = 'D:\\Server_Apps\\7zip\\7za.exe a'

-- Set this to the path and name of your remove folder command, including switches
-- for it to be quiet (used with physics-related images)
local rmdir_cmd = 'rmdir /s/q'
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------
-- User-defined lists that determine how Orthanc deals with certain studies

-- A list to check against patient name and ID to see if the images should be kept.
-- Orthanc will put anything that matches this in the physics_to_keep_folder.
local physics_to_keep = {'physics'}

-- Lists of things to ignore. Orthanc will ignore anything matching the content of
-- these lists: they will not be imported into OpenREM.
local manufacturers_to_ignore = {'Agfa', 'Agfa-Gevaert', 'Agfa-Gevaert AG', 'Faxitron X-Ray LLC', 'Gendex-KaVo'
local model_names_to_ignore = {'CR 85', 'CR 75', 'CR 35', 'CR 25', 'ADC_5146', 'CR975'}
local station_names_to_ignore = {'CR85 Main', 'CR75 Main'}
```

```
-- Set this to the path where you want to keep physics-related DICOM images
local physics_to_keep_folder = 'E:\\dicom\\physics\\'

-- Set this to the path and name of your zip utility (used with physics-related images)
local zip_executable = 'D:\\Server_Apps\\7zip\\7za.exe a'

-- Set this to the path and name of your remove folder command, including switches
-- for it to be quiet (used with physics-related images)
local rmdir_cmd = 'rmdir /s/q'
--------------------------------------------------------------------------------



--------------------------------------------------------------------------------
-- User-defined lists that determine how Orthanc deals with certain studies

-- A list to check against patient name and ID to see if the images should be kept.
-- Orthanc will put anything that matches this in the physics_to_keep_folder.
local physics_to_keep = {'physics'}

-- Lists of things to ignore. Orthanc will ignore anything matching the content of
-- these lists: they will not be imported into OpenREM.
local manufacturers_to_ignore = {'Agfa', 'Agfa-Gevaert', 'Agfa-Gevaert AG', 'Faxitron X-Ray LLC', 'Gendex-KaVo'}
local model_names_to_ignore = {'CR 85', 'CR 75', 'CR 35', 'CR 25', 'ADC_5146', 'CR975'}
local station_names_to_ignore = {'CR85 Main', 'CR75 Main'}
local software_versions_to_ignore = {'VixWin Platinum v3.3'}
local device_serial_numbers_to_ignore = {}
local rdsr_serial_numbers_to_ignore = {'SCB1312016'}

-- Set this to true if you want to use the OpenREM Toshiba CT extractor. Set it to
-- false to disable this feature.
local use_toshiba_ct_extractor = true

-- A list of CT make and model pairs that are known to have worked with the Toshiba CT extractor
local toshiba_extractor_systems = {
        {'GE Medical Systems', 'Discovery 710'},
        {'GE Medical Systems', 'Discovery STE'},
        {'GE Medical Systems', 'Brightspeed'},
        {'GE Medical Systems', 'Lightspeed Plus'},
        {'GE Medical Systems', 'Lightspeed16'},
        {'GE Medical Systems', 'Lightspeed Pro 32'},
        {'GE Medical Systems', 'Lightspeed VCT'},
        {'Siemens', 'Biograph64'},
        {'Siemens', 'Somatom Definition'},
        {'Siemens', 'Somatom Definition Edge'},
        {'Siemens', 'Somatom Definition Flash'},
```

```lua
function OnStoredInstance(instanceId)
    -- Retrieve the DICOM tags from the instance. The tags parameter doesn't include all the useful
    -- tags - this does.
    local instance_tags = ParseJson(RestApiGet('/instances/' .. instanceId .. '/simplified-tags'))

    ----------------------------------------------------------------------------------------
    -- See if the images are physics tests - if so, keep them and exit this function
    if use_physics_filtering == true then

        local patient_name = 'blank'
        local patient_id = 'blank'

        if (instance_tags.PatientName ~= nil and instance_tags.PatientName ~= '') then
            patient_name = instance_tags.PatientName
        else
            patient_name = 'blank'
        end

        if (instance_tags.PatientID ~= nil and instance_tags.PatientID ~= '') then
            patient_id = instance_tags.PatientID
        elseif (instance_tags.RETIRED_OtherPatientIDs ~= nil and instance_tags.RETIRED_OtherPatientIDs ~= '') th
            patient_id = instance_tags.RETIRED_OtherPatientIDs
        else
            patient_id = 'blank'
        end

        for i = 1, #physics_to_keep do
            if string.match(string.lower(patient_name), string.lower(physics_to_keep[i])) then
                print('It is a physics test. PatientName is: ' .. patient_name)
                print('Sending this to the medphysics-11 Orthanc node')
                SendToModality(instanceId, 'OrthancMedphys')
                return true
            end

            if string.match(string.lower(patient_id), string.lower(physics_to_keep[i])) then
                print('It is a physics test. PatientID is: ' .. patient_id)
                print('Sending this to the medphysics-11 Orthanc node')
                SendToModality(instanceId, 'OrthancMedphys')
                return true
            end
        end

    end
    ----------------------------------------------------------------------------------------
```

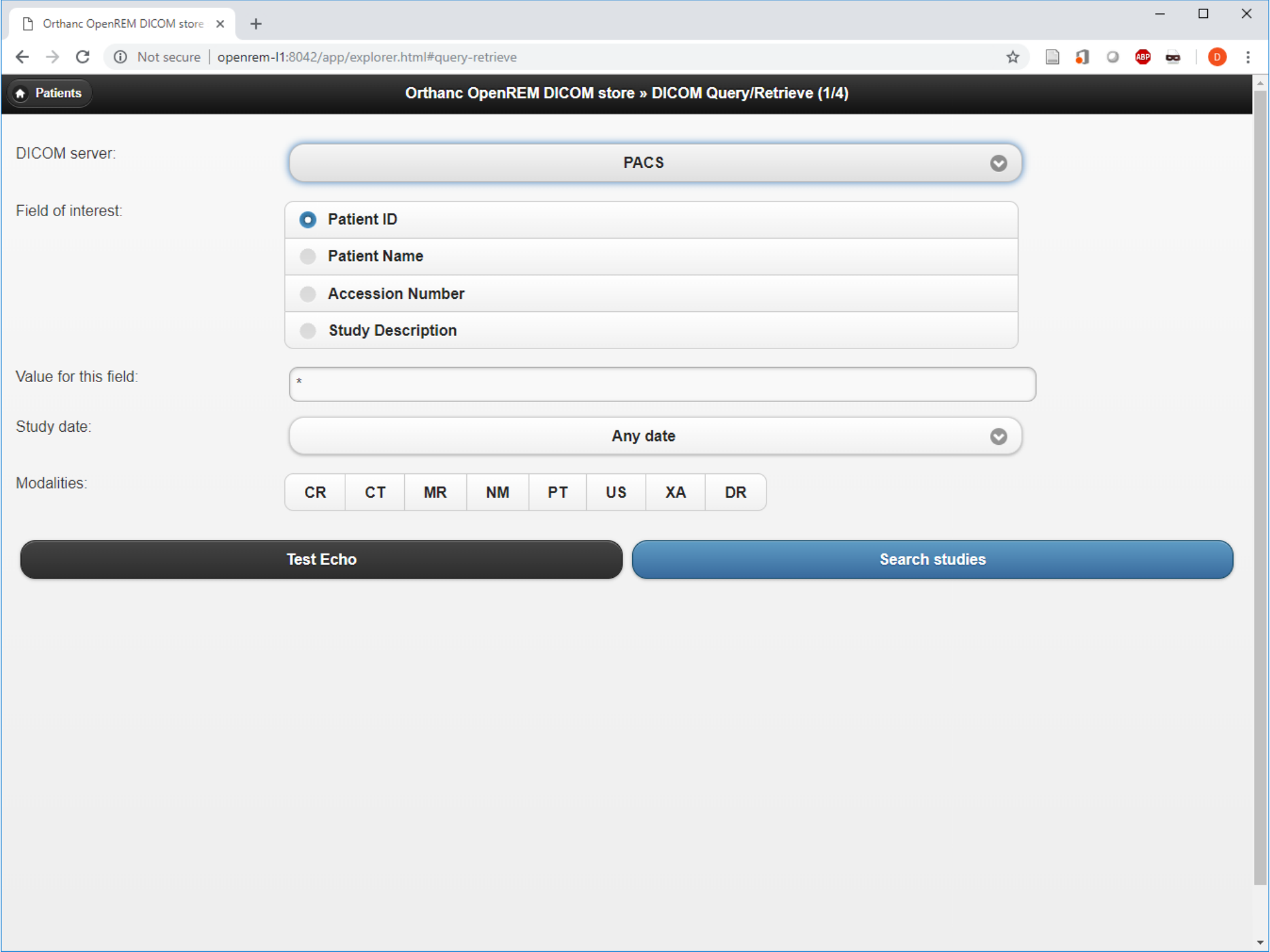
```lua
function OnStableStudy(studyId)
    -- Retrieve the shared DICOM tags from the study. The tags parameter doesn't include
    -- all the useful tags - this does
    local study_tags = ParseJson(RestApiGet('/studies/' .. studyId .. '/shared-tags?simplify'))

    ----------------------------------------------------------------------------------------
    -- See if any of the physics strings are in patient name or ID. If they are then
    -- copy the image to the physics_to_keep_folder and then remove it from Orthanc
    if use_physics_filtering == true then
        local patient_name = 'blank'
        local patient_id = 'blank'
        local patient_folder = 'blank'
        if (study_tags.PatientName ~= nil and study_tags.PatientName ~= '') then
            patient_name = study_tags.PatientName
            patient_folder = patient_name
        else
            patient_name = 'blank'
        end
        if (study_tags.PatientID ~= nil and study_tags.PatientID ~= '') then
            patient_id = study_tags.PatientID
            if patient_folder == 'blank' then
                patient_folder = patient_id
            end
        elseif (study_tags.RETIRED_OtherPatientIDs ~= nil and study_tags.RETIRED_OtherPatientIDs ~= '') then
            patient_id = study_tags.RETIRED_OtherPatientIDs
            if patient_folder == 'blank' then
                patient_folder = patient_id
            end
        else
            patient_id = 'blank'
        end

        for i = 1, #physics_to_keep do
            if string.match(string.lower(patient_name), string.lower(physics_to_keep[i])) or string.match(string
                -- It is a physics patient - save them to the physics folder
                print('It is physics')
                local first_series = true
                local temp_files_path = ''

                -- Retrieve the IDs of all the series in this study
                local series = ParseJson(RestApiGet('/studies/' .. studyId)) ['Series']

                -- using _ as a placeholder as I'm not interested in the key value
                for _, current_series in pairs(series) do
```

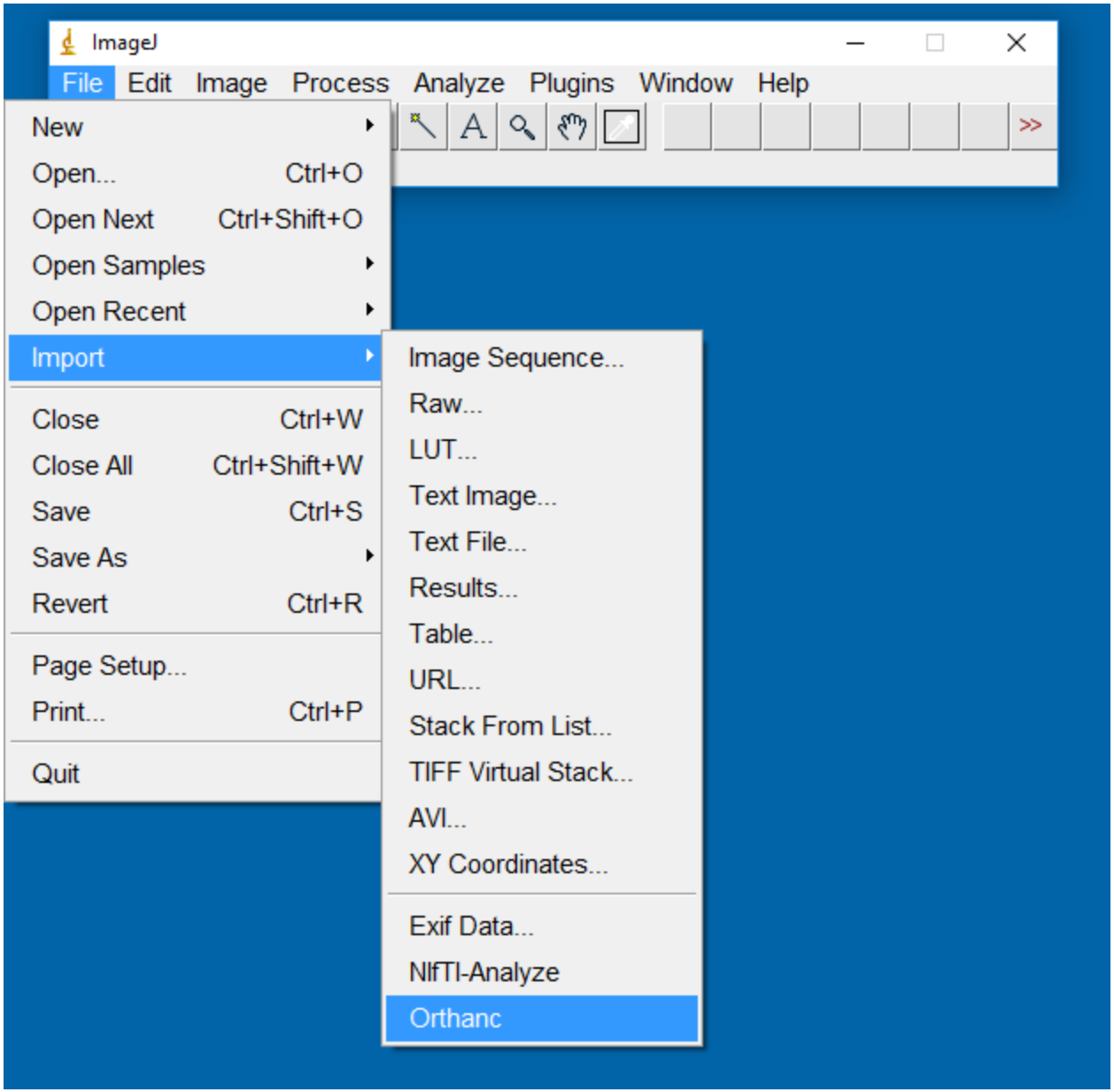
```
358
359             -- using _ as a placeholder as I'm not interested in the key value
360             for _, current_series in pairs(series) do
361
362                 if first_series == true then
363                     -- Create a string containing the folder path.
364                     temp_files_path = ToAscii(physics_to_keep_folder .. study_tags.StudyDate .. dir_sep .. p
365
366                     -- Create the folder
367                     os.execute(mkdir_cmd .. ' "' .. temp_files_path .. '"')
368
369                     first_series = false
370                 end
371
372                 local instances = ParseJson(RestApiGet('/series/' .. current_series)) ['Instances']
373
374                 -- Loop through each instance in the current_series
375                 -- using _ as a placeholder as I'm not interested in the key value
376                 for _, instance in pairs(instances) do
377                     -- Retrieve the DICOM file from Orthanc
378                     local dicom = RestApiGet('/instances/' .. instance .. '/file')
379
380                     -- Write the DICOM file to the folder created earlier
381                     local target = assert(io.open(temp_files_path .. dir_sep .. instance .. '.dcm', 'wb'))
382                     target:write(dicom)
383                     target:close()
384
385                     -- Remove the instance from Orthanc
386                     Delete(instance)
387                 end
388             end
389
390             -- Zip the study files to save space and remove the originals after zipping
391             print('Zipping physics images: ' .. zip_executable .. ' "' .. temp_files_path .. '.zip"' .. ' "
392             os.execute(zip_executable .. ' "' .. temp_files_path .. '.zip"' .. ' "' .. temp_files_path .. di
393             print('Removing physics study folder: ' .. rmdir_cmd .. ' "' .. temp_files_path .. '"')
394             os.execute(rmdir_cmd .. ' "' .. temp_files_path .. '"')
395
396             -- Exit the function, as a physics study was found and the images moved
397             return true
398
399         end
400     end
401 end
402 ----------------------------------------------------------------------------------
```

A simpler approach?

A simpler approach?

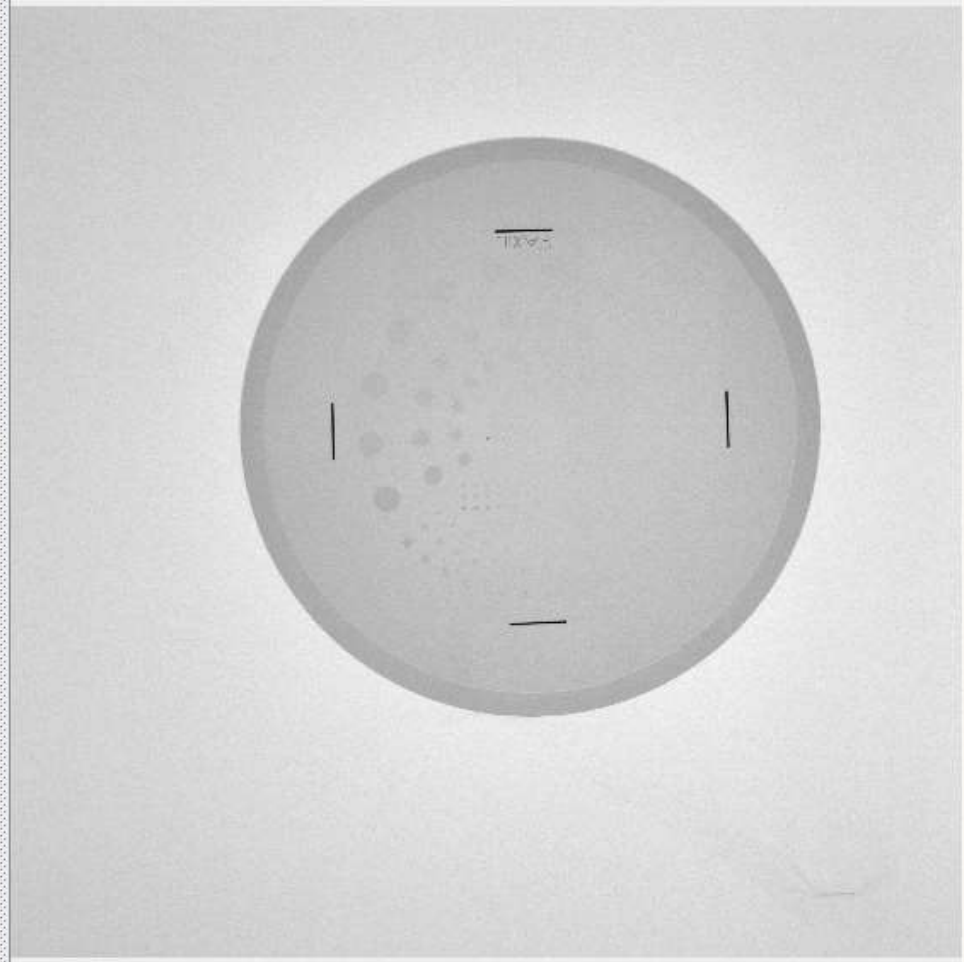
- Install Orthanc on a computer that has a fixed IP address
  - Ask your IT department
- Ask your PACS manager to give your Orthanc server permission to query / retrieve PACS
  - You'll need to know the IP address, AE Title and port number of your Orthanc install
- Use the web interface of Orthanc to transfer QA images from PACS to your Orthanc server
- You can use ImageJ on any networked computer to open images that are on the Orthanc server


Orthanc OpenREM DICOM store
Not secure | openrem-l1:8042/app/explorer.html#query-retrieve
Patients
Orthanc OpenREM DICOM store » DICOM Query/Retrieve (1/4)
DICOM server:
PACS
Field of interest:
Patient ID
Patient Name
Accession Number
Study Description
Value for this field:
*
Study date:
Any date
Modalities:
CR
CT
MR
NM
PT
US
XA
DR
Test Echo
Search studies


ImageJ
File
Edit
Image
Process
Analyze
Plugins
Window
Help
New
Open... Ctrl+O
Open Next Ctrl+Shift+O
Open Samples
Open Recent
Import
Close Ctrl+W
Close All Ctrl+Shift+W
Save Ctrl+S
Save As
Revert Ctrl+R
Page Setup...
Print... Ctrl+P
Quit
Image Sequence...
Raw...
LUT...
Text Image...
Text File...
Results...
Table...
URL...
Stack From List...
TIFF Virtual Stack...
AVI...
XY Coordinates...
Exif Data...
NIfTI-Analyze
Orthanc

Select some series or some instance in Orthanc

- OrthancNewton
- OrthancMedphys
  - 38375673 - PHYSICS1809^AECSTEREO
  - 595967867 - PHYSICS1809^TORMAM
  - 909878676 - PHYSICS1809^TORMAM
  - PHYSICS C FIXED TEST^PHYSICS C FIXED TE
    - ------ 20180909
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - W Image Quality Vertical Plate
      - CR - X Tube Output
    - ------ 20180909
    - 20180629
    - 20180629
    - 20180710
    - 20180717
    - 20180718
    - 20180718
    - 20180728
    - 20180728
    - 20180807
    - 20180810
    - 20180810
    - 20180810
    - 20180901
    - 20180906
    - 20180906
    - 20180906
    - 20180906
    - SYSTEM DIAGNOSIS - 20180901



Add server | Remove server | Open | Cancel

Summary

If you have OpenREM	If no OpenREM
May already have Orthanc	Obtain a “server”, install Orthanc
May already have permission to query PACS	Obtain permission to query PACS
Edit Orthanc Lua script using instructions in OpenREM docs and enable physics filtering	-
Automatic filtering will then occur whenever DICOM objects arrive at OpenREM (manual query needed to obtain all CT images)	Manually query PACS as and when you want to retrieve images
ImageJ Orthanc plugin can be used to browse and load images	
Images can be saved to the file system from Orthanc webpage	

Thanks for listening

Any questions?