XNAT Desktop: communicating non-DICOM data with XNAT

A Field Guide

XNAT

Contents

- XNAT Desktop: what, where, why
- Uploading files to XNAT
- File tagging techniques in XND
- Experimental features and future development
- Use of REST API by XND in communications with XNAT

XNAT

How do I upload non-DICOM data to XNAT?

- Manual
- Automatic script
- XNAT Desktop (XND)

XND: what

- Cross platform
- Graphical user interface
- File-centric application
- Source code: Java+Eclipse Rich Client
- Beta version

Attribute-value (tag) notion in XND

- Main XND objects: tags and files
- A tag is a <Name-Value> pair
- [tag1, tag2, ..] <-> individual file

XNAT

Required tags to upload to XNAT

- XNAT container hierarchy:
  Project << Subject << Experiment << [Scan, Assessment, Reconstruction]
- Modality: MR, CT, PET, US, ..
- Scan_Type: DICOM, TIFF, ...: required for XNAT image viewer
- (Optional): Quality, Date

XND Browsing

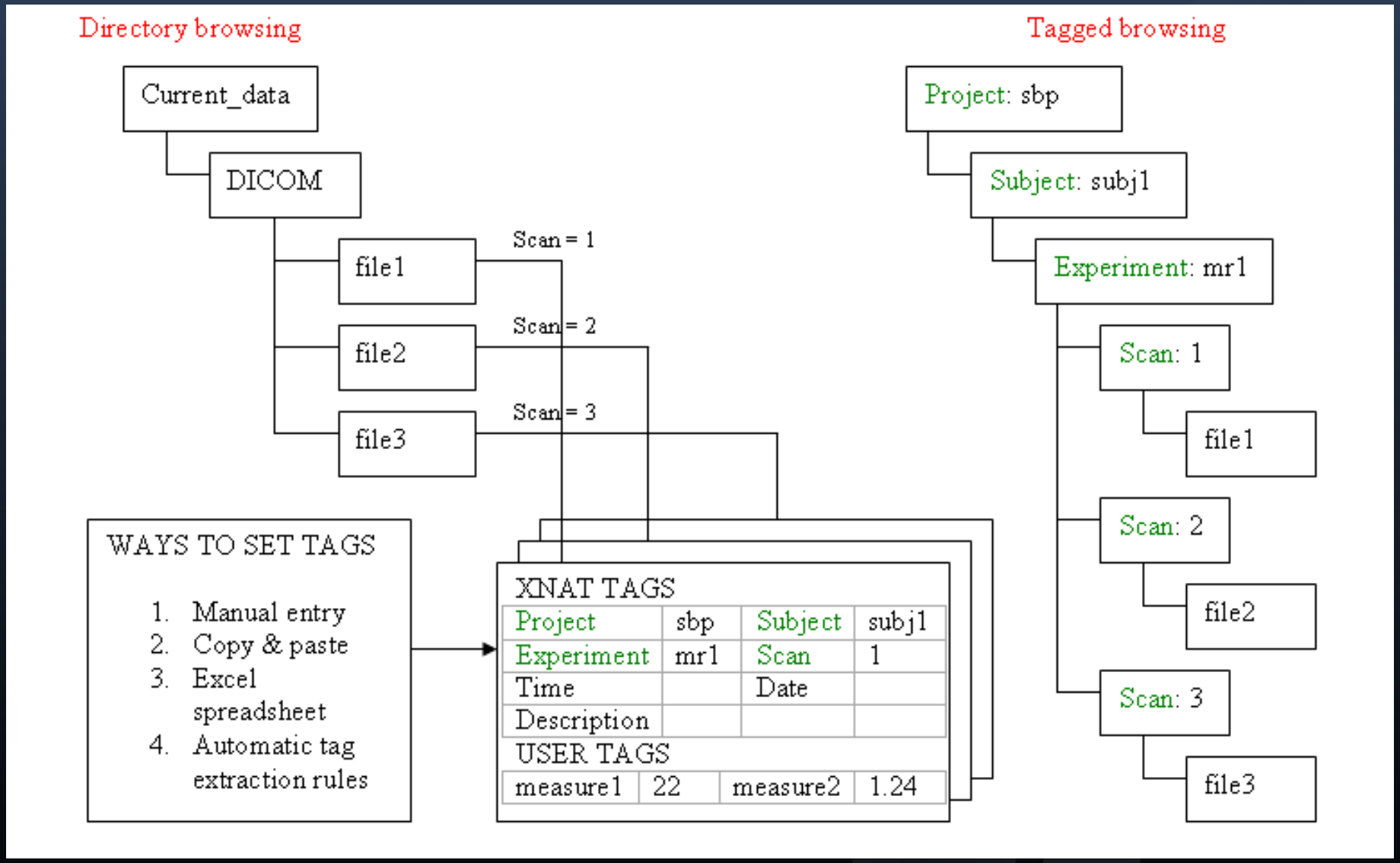
- Going through various file containers
- Manage/unmanage files
- Assigning tags to files within containers

Folder containers in XND: File System

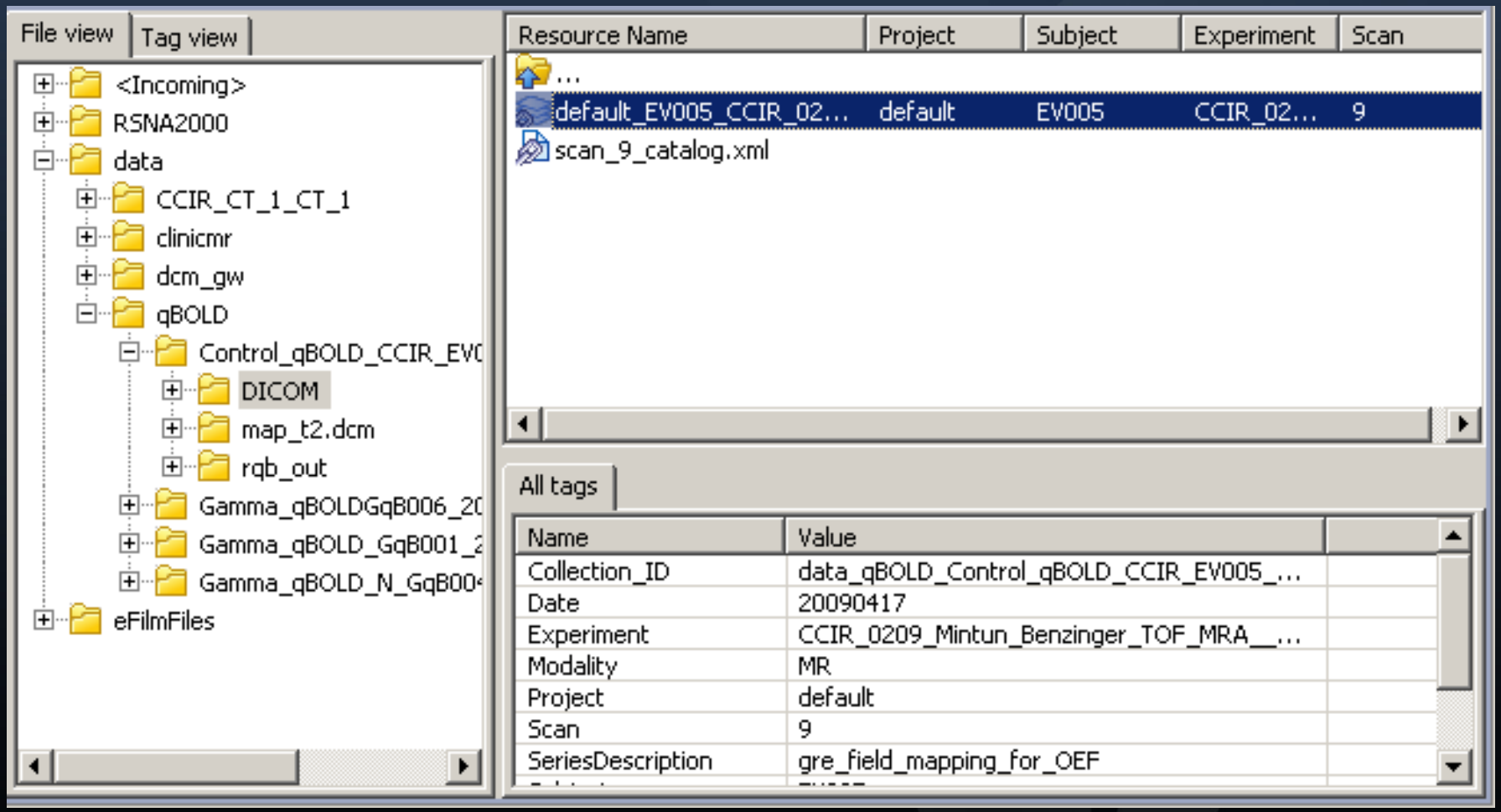
- Managed root
- Special: <Incoming> folder
- Subfolders under managed root

Non-FS file containers

- Virtual Folder (VF): certain tags can be assigned to show as VF's in hierarchy
- Collection (~XNAT resource): all files within collection share the same tags.

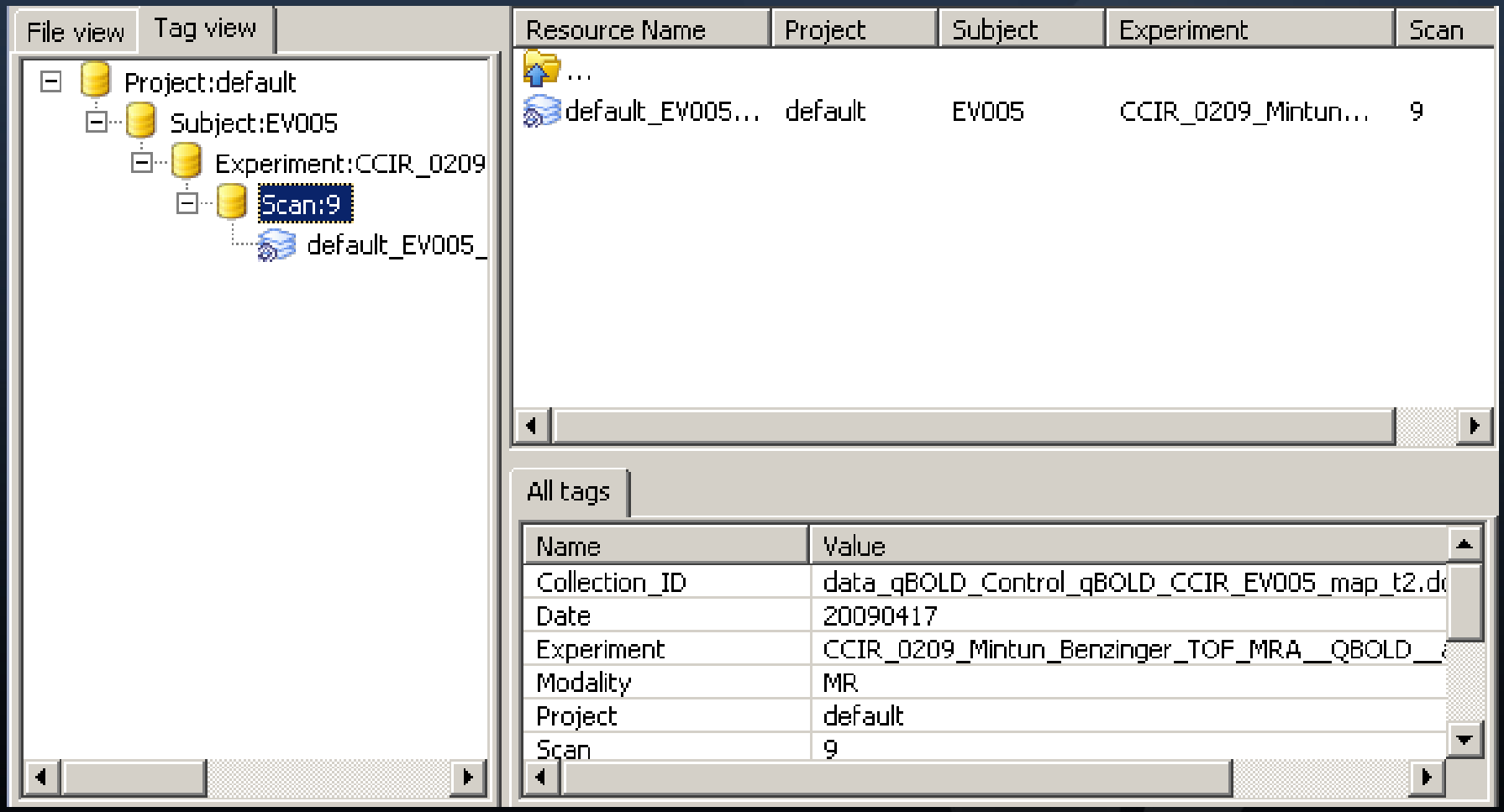

Directory browsing
Tagged browsing
Current_data
DICOM
file1
file2
file3
Scan = 1
Scan = 2
Scan = 3
WAYS TO SET TAGS
1. Manual entry
2. Copy & paste
3. Excel spreadsheet
4. Automatic tag extraction rules
XNAT TAGS
Project sbp Subject subj1
Experiment mr1 Scan 1
Time Date
Description
USER TAGS
measure1 22 measure2 1.24
Project: sbp
Subject: subj1
Experiment: mr1
Scan: 1
file1
Scan: 2
file2
Scan: 3
file3

XND folder view



XNAT

XND tag view


File view
Tag view
Project:default
Subject:EV005
Experiment:CCIR_0209
Scan:9
default_EV005_
Resource Name
Project
Subject
Experiment
Scan
...
default_EV005...
default
EV005
CCIR_0209_Mintun...
9
All tags
Name
Value
Collection_ID
data_qBOLD_Control_qBOLD_CCIR_EV005_map_t2.d
Date
20090417
Experiment
CCIR_0209_Mintun_Benzinger_TOF_MRA__QBOLD__
Modality
MR
Project
default
Scan
9

Upload from XND to XNAT

- Use Tag View to upload
- Can upload files at any level
- Single/multiple container upload

(Upload demo)

Steps to tag files in XND

- Add managed root (rule of thumb: use lowest level directory possible)
- Manage some files in subfolders
- Select a container
- Apply tagging procedure to the container
- Repeat using tag copy/paste feature

Manual tagging example (see the demo)

- Manual: set ‘Project’ and ‘quality’
- Manual: add Experiment and Scan values to one of subfolders
- Use clipboard copy/paste to quickly populate tags

Sources of information about local files

- File contents
- File names & extensions
- Directory path
- External sources (spreadsheets, xml, etc.)

Automatic tagging: rules

- Metadata source
- Default logic (built-in),
- Custom logic (user-defined in xml file using rule language)

Automatic tagging: DICOM rule

- Source: DICOM tags
- Default logic: create collections based on matching tag hierarchies
- Custom logic: default (XNAT-specific) or user-supplied tag map

DICOM rule tag matching

- (element, group) <-> XND tag
- Default value
- Assignment prioritization

```
<tag name="Subject" defaultValue="subj_undefined">
  <DICOMTag group="0010" element="0010" alias="PatientName" priority="1" />
  <DICOMTag group="0010" element="0020" alias="PatientID" priority="2" />
</tag>
```

XNAT

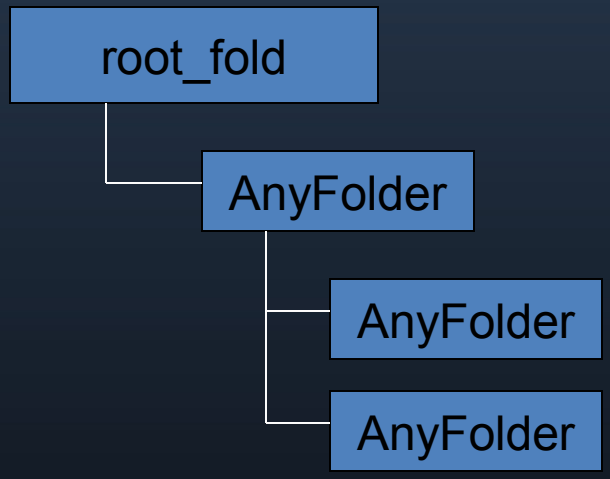
(DICOM rule demo)

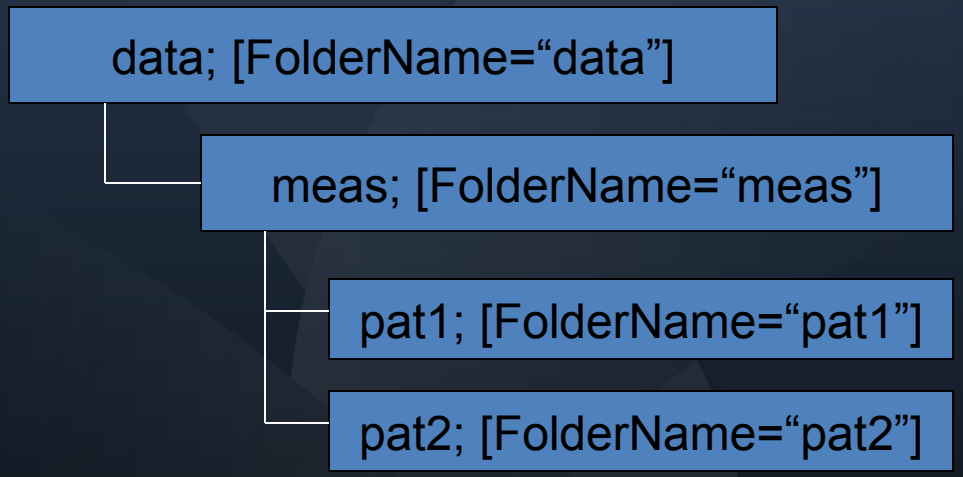
XNAT

Automatic tagging: naming rule

- Metadata source: folder structure/file name pattern
- Default logic: recursive processing of selected folder hierarchy
- Custom logic: match parts of file/folder names with regular expressions

Naming rule: modeling folder structure

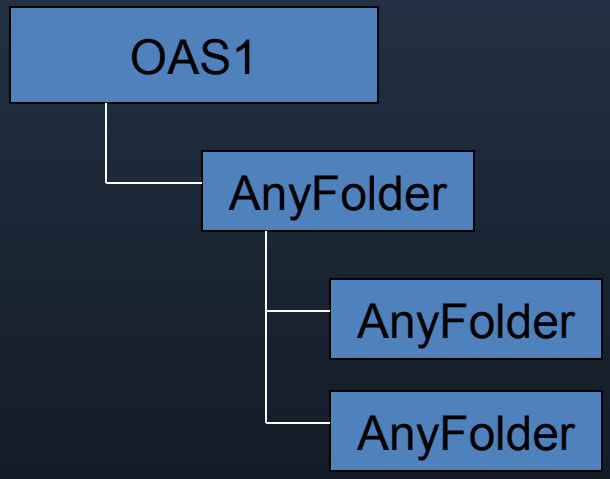


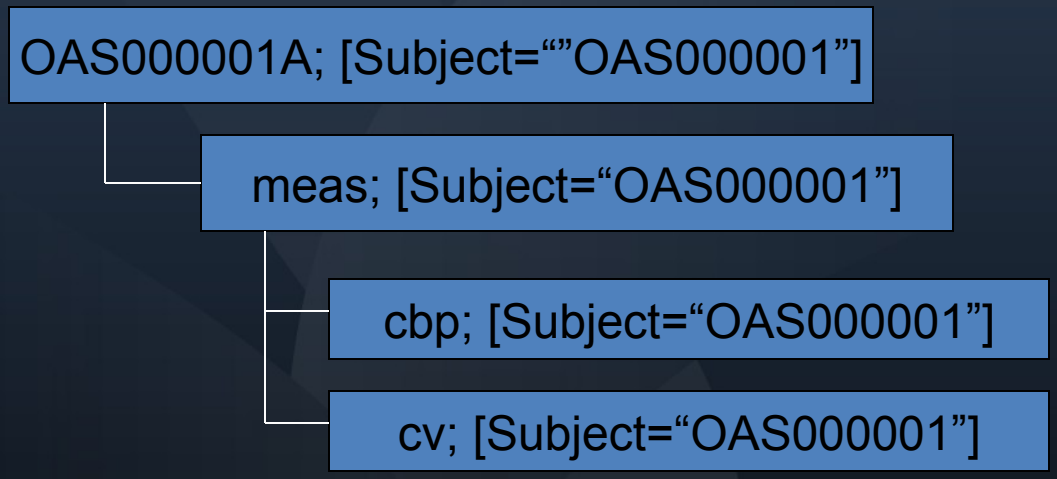



```
<folderID="root_fold" pattern=".*" treeRoot="1">
  <child>AnyFolder</child>
  <child>label</child>
</folder>
<folderID="AnyFolder" pattern=".*">
  <tag name="FolderName" pattern=".*" recursive="pattern" />
  <child>AnyFolder</child>
</folder>
```

XNAT

Naming rule: capturing folder name patterns

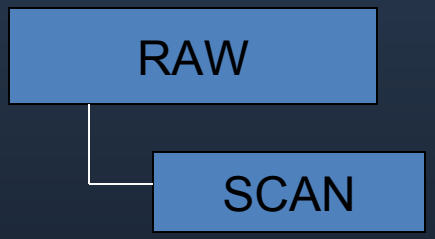


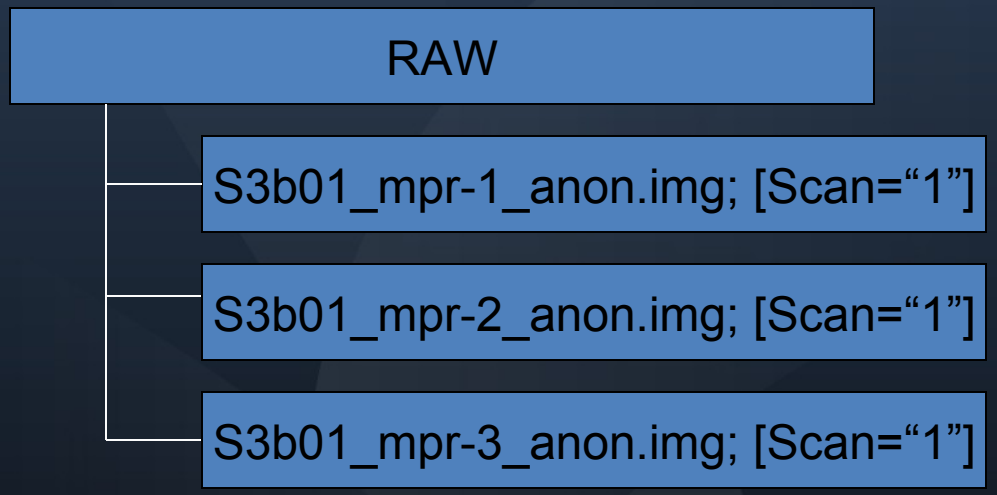



```
<folderID="OAS1" pattern="OAS1.*" treeRoot="1">
  <tag name="Subject" pattern="(.{9}).* " recursive="fixed" />
</folder>
<folderID="FSL_SEG">
```

XNAT

Naming rule: file name patterns





Folder ID =
Folder Name

```
<folderID="RAW">
  <child>SCAN</child>
</folder>
<fileID="SCAN" pattern=".*">
  <tag name="Scan" pattern=".*_mpr-(.)_anon.* "
    recursive="pattern" />
</file>
```

XNAT

(Naming rule demo)

XNAT

Automatic tagging, tag pattern rule

- Source: existing tags in XND
- Default logic: use specified tag to derive a modified value
- Custom logic: user-defined tag map (regexp based)

Tag pattern: formatting subject name example

```
<tag name="Subject">
  <substring pattern=".{4}(.{3})*" value="0{1}"/>
</tag>
<tag name="Scan_Type">
  <replace match="[\x3c\x5b\x3e\x5d]" with=""/>
  <replace match="[ ]" with="_"/>
  <replace match="[-]" with="_"/>
  <replace match="[\x26]" with="_"/>
</tag>
```

file1 [Subject="wcaf01", Scan_Type="pre-dti.2mm&low^contrast"

file1 [Subject="001", Scan_Type="pre_dti_.2mm_low_contrast"

XNAT

Macro: a tool for one-click data markup

- Macro is a sequence of commands of three following types:
  - Manage
  - Apply rule
  - Set a tag

- Construct macros for re-occurring data patterns for quick tag & upload.

XNAT

(Macro demo)

XNAT

Conclusions

- Use XNAT Desktop for uploading multi-format data to XNAT
- Use supplied tagging rules or create your own
- Tagging procedure can, with some customization, be made automatic in most cases

What's next? (Features in development. Early previews available)

- Data import wizard
- DICOM networking capabilities: import directly from PACS
- Research PACS workstation interface: extended DICOM image markup and viewing

XNAT

XND: building REST queries

- XNAT:
  Project << Subject << Experiment << [Scan, Assessment, Reconstruction

- XND:

<Project="default"><Subject="EV005"><Experiment="CCIR00209.."><Scan="9">

XNAT

Tag relationships in XND

```
<ontology id="xnat" descr="Default ontology with XNAT hierarchical tags.">
 <tag name="Project" treeRoot="1" tableView="1" anValue="1" type="ontology">
  <child>Subject</child>
  <child>Resource</child>
 </tag>
 <tag name="Subject" tableView="1" anValue="1" type="ontology" context="1">
   <child>Experiment</child>
   <child>Resource</child>
 </tag>
 <tag name="Experiment" tableView="1" anValue="1" type="ontology">
   <child>Scan</child>
   <child>Assessor</child>
   <child>Reconstruction</child>
   <child>Resource</child>
 </tag>
```

XNAT

XND <-> XNAT communication

File: meas1.xml; Project="oawbm", Subject="003"

XND tag engine

XND REST adapter/ http client

XND/XNAT table parser

http://myxnatserver/REST/ Projects/oawbm/Subjects/003

XNAT server

sub_project_identifier_default	subjectid	projects
Abdomen_Kidney	CNINDS05L_S00032 ,	
ABDOMEN_VOLUNTEER	CNINDS05L_S00027 ,	
AORTA	CNINDS05L_S00023 ,	
AUTOM_INTERACTIVE_BC_plus	CNINDS05L_S00024 ,	
AVE__ABDOMEN	CNINDS05L_S00025 ,	
AVE__BRAIN___POST_FOS.	CNINDS05L_S00026 ,	

XNAT

XND upload strategy

- Determine level at which upload should be performed.
- Execute GET query at each hierarchy level. Find lowest non-existent level.
- Re-create hierarchy using PUT methods that create XNAT objects
- Upload under created hierarchy.
- Repeat for multiple items

REST upload example

GET (success): http://central.xnat.org/REST/**experiments?xsiType=xnat:subjectAssessorData&project=default**&format=xml
GET (error code 404): http://central.xnat.org/REST/**projects/default/subjects/EV005/experiments/CCIR_0209/scans/9**/?format=xml
GET (error code 404): http://central.xnat.org/REST/**projects/default/subjects/EV005/experiments/CCIR_0209**/?format=xml
GET (success): http://central.xnat.org/REST/**projects/default/subjects/EV005**/?format=xml
PUT (success): http://central.xnat.org/REST/**projects/default/subjects/EV005/experiments/CCIR_0209**/?date=20090417&label=CCIR_0209&xsiType=xnat:MRSessionData&format=xml
PUT (success): http://central.xnat.org/REST/**projects/default/subjects/EV005/experiments/CCIR_0209/scans/9**/?ID=9&series_description=gre_field_mapping_for_OEF&format=xml
PUT (success): http://central.xnat.org/REST/**projects/default/subjects/EV005/experiments/CCIR_0209/scans/9/resources/DICOM**/?content=RAW&format=DICOM&
PUT (success): http://central.xnat.org/REST/**projects/default/subjects/EV005/experiments/CCIR_0209/scans/9/resources/DICOM/files/files.zip**?inbody=true&extract=true&format=DICOM&content=RAW&
PUT (success): http://central.xnat.org/REST/**projects/default/subjects/EV005/experiments/CCIR_0209**/?pullDataFromHeaders=true&format=xml
PUT (success): http://central.xnat.org/REST/**projects/default/subjects/EV005/experiments/CCIR_0209**/?triggerPipelines=true&format=xml

Resources

- **XND manual:** http://www.xnat.org/xnd/
- XND download: http://nrg.wustl.edu/xnd/download/
- Java regular expression specification: http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html
- Regular expression test applet: http://www.cis.upenn.edu/~matuszek/General/RegexTester/regex-tester.html