# The TCIA Plugin and Servlet API

## Contents


1 Installing the TCIA Software ..... 3
2 The config.xml file ..... 3
3 User Accounts ..... 6
4 The TCIAServlet API ..... 7
4.1 Submitting Files to the Import Pipeline ..... 7
4.1.1 Listing Files on the Server ..... 7
4.1.2 Checking the Space Required for Import ..... 7
4.1.3 Submitting Files to the DirectoryImportService ..... 8
4.1.4 Submitting Files to the DicomImportService ..... 8
4.2 Viewing the Status of the Import Pipeline ..... 8
4.3 The Import Manifest ..... 8
4.3.1 Listing the Import Manifest as XML ..... 8
4.3.2 Listing the Import Manifest as CSV ..... 9
4.3.3 Listing the Import Manifest as XLSX ..... 9
4.4 Updating the DicomAnonymizer Lookup Table ..... 9
4.4.1 Listing the Lookup Table Template Spreadsheet ..... 9
4.4.2 Submitting the Lookup Table ..... 10
4.5 Viewing the List of Files Ready for Anonymization ..... 10
4.6 Initializing the Anonymization Pipeline Counts ..... 11
4.7 Submitting Files for Anonymization ..... 12
4.8 Pausing the Anonymization Pipeline ..... 12
4.9 Restarting a Paused Anonymization Pipeline ..... 12
4.10 Viewing the List of Files Ready for Export ..... 12
4.11 Exporting Files ..... 13
4.12 The Export Manifest Database ..... 13
4.12.1 Clearing the Export Manifest Database ..... 14
4.12.2 Getting the Export Manifest Database Status ..... 14
4.12.3 The Local Manifest ..... 14
4.12.3.1 Listing the Manifest with PHI as XML ..... 14

4.12.3.2 Listing the Manifest with PHI as CSV ..... 14
4.12.3.3 Listing the Manifest with PHI as XLSX ..... 15
4.12.4 The Export Manifest ..... 15
4.12.4.1 Listing the Manifest without PHI as XML ..... 15
4.12.4.2 Listing the Manifest without PHI as CSV ..... 15
4.12.4.3 Listing the Manifest without PHI as XLSX ..... 15
4.12.5 Exporting the Manifest ..... 16
4.13 Image Functions ..... 16
4.13.1 Listing the Elements of a DICOM File ..... 16
4.13.2 Getting a JPEG Image from a DICOM File ..... 16
4.14 Special Functions ..... 16
4.14.1 Getting the List of File System Roots on the Server ..... 16
4.14.2 Getting the Available Space on the Server ..... 17
4.14.3 Getting the Export Queue Size ..... 17
4.14.4 Getting the URL of the Quarantine Manager Servlet ..... 17
4.14.5 Getting a Summary of the DicomAnonymizer Quarantine ..... 17
4.14.6 Shutting Down the Server ..... 17
4.15 Testing Functions ..... 18

# 1 Installing the TCIA Software

The best way to install the TCIA software is to run the TCIA-installer.jar program. It runs exactly like the CTP-installer program and can be used to perform a fresh installation or upgrade an existing one.

**IMPORTANT**: The installer never overwrites an existing config.xml file, so when upgrading an installation to a new version that has a different configuration, delete (or rename) the old config.xml before running the installer. This will allow the installer to put the new version in place.

The installer installs:

1. CTP
2. config.xml
3. The TCIAPlugin library
4. The wizard home page
5. The wizard libraries

# 2 The config.xml file

The config.xml for the TCIA project is:

```
<Configuration>
    <Server
        maxThreads="20"
        port="9000"/>
    <Plugin
        anonymizerID="CollectionDicomAnonymizer"
        anonymizerInputID="CollectionAnonymizerInput"
        anonymizerStorageID="CollectionAnonymizerStorage"
        class="edu.uams.tcia.TCIAPlugin"
        exportInputID="CollectionExportInput"
        exportManifestLogID="CollectionExportManifestLog"
        exportOutputID="CollectionExportOutput"
        id="Collection"
        importInputID="CollectionImportInput"
        importManifestLogID="CollectionImportManifestLog"
        importStorageID="CollectionImportStorage"
        name="CollectionTCIAPlugin"
        root="Collection/TCIAplugin"/>
    <Plugin
        class="edu.uams.tcia.ExportManifestLogPlugin"
        id="CollectionExportManifestLog"
        name="CollectionExportManifestLog"
        root="Collection/ExportManifestLog"
        tciaPluginID="Collection"/>
    <Plugin
        class="edu.uams.tcia.ImportManifestLogPlugin"
        id="CollectionImportManifestLog"
        name="CollectionImportManifestLog"
        root="Collection/ImportManifestLog"/>
    <Pipeline
        name="CollectionImport"
        root="Collection/import">
```

```
        <DirectoryImportService
            class="org.rsna.ctp.stdstages.DirectoryImportService"
            id="CollectionImportInput"
            import="roots/DirectoryImportService/import"
            interval="4000"
            name="DirectoryImportService"
            quarantine="quarantines/DirectoryImportService"
            root="roots/DirectoryImportService"/>
        <DicomImportService
            class="org.rsna.ctp.stdstages.DicomImportService"
            logConnections="no"
            name="DicomImportService"
            port="104"
            quarantine="quarantines/DicomImportService"
            root="roots/DicomImportService"/>
        <ImportManifestLogger
            class="edu.uams.tcia.ImportManifestLogger"
            id="CollectionImportManifestLogger"
            manifestLogID="CollectionImportManifestLog"
            name="ImportManifestLogger"
            root="roots/CollectionImportManifestLogger"/>
        <DirectoryStorageService
            acceptDuplicates="yes"
            class="org.rsna.ctp.stdstages.DirectoryStorageService"
            defaultString=""
            id="CollectionImportStorage"
            logDuplicates="no"
            name="DirectoryStorageService"
            quarantine="quarantines/DirectoryStorageService"
            root="roots/DirectoryStorageService"
            setStandardExtensions="no"
            structure="(0010,0020)-(0010,0010)/(0008,0020)/Series(0020,0011)"
            whitespaceReplacement="_"/>
    </Pipeline>
    <Pipeline
        name="CollectionAnonymization"
        root="Collection/anonymization">
        <DirectoryImportService
            class="org.rsna.ctp.stdstages.DirectoryImportService"
            id="CollectionAnonymizerInput"
            import="roots/DirectoryImportService/import"
            interval="4000"
            name="DirectoryImportService"
            quarantine="quarantines/DirectoryImportService"
            root="roots/DirectoryImportService"/>
        <ObjectCache
            class="org.rsna.ctp.stdstages.ObjectCache"
            id="CollectionObjectCache"
            name="ObjectCache"
            root="roots/ObjectCache"/>
        <DicomAnonymizer
            class="org.rsna.ctp.stdstages.DicomAnonymizer"
            id="CollectionDicomAnonymizer"
            lookupTable="scripts/LookupTable.properties"
            name="DicomAnonymizer"
            quarantine="quarantines/DicomAnonymizer"
            root="roots/DicomAnonymizer"
```

```
            script="scripts/TCIADicomAnonymizer.script"/>
        <ExportManifestLogger
            cacheID="CollectionObjectCache"
            class="edu.uams.tcia.ExportManifestLogger"
            id="CollectionExportManifestLogger"
            manifestLogID="CollectionExportManifestLog"
            name="ExportManifestLogger"
            root="roots/CollectionExportManifestLogger"/>
        <DirectoryStorageService
            acceptDuplicates="yes"
            cacheID="CollectionObjectCache"
            class="org.rsna.ctp.stdstages.DirectoryStorageService"
            defaultString=""
            id="CollectionAnonymizerStorage"
            logDuplicates="no"
            name="DirectoryStorageService"
            quarantine="quarantines/DirectoryStorageService"
            root="roots/DirectoryStorageService"
            setStandardExtensions="no"
            structure="(0010,0020)-(0010,0010)/(0008,0020)/Series(0020,0011)"
            whitespaceReplacement="_"/>
    </Pipeline>
    <Pipeline
        name="CollectionExport"
        root="Collection/export">
        <DirectoryImportService
            class="org.rsna.ctp.stdstages.DirectoryImportService"
            id="CollectionExportInput"
            import="roots/DirectoryImportService/import"
            interval="4000"
            name="DirectoryImportService"
            quarantine="quarantines/DirectoryImportService"
            root="roots/DirectoryImportService"/>
        <HttpExportService
            class="org.rsna.ctp.stdstages.HttpExportService"
            id="CollectionExportOutput"
            name="HttpExportService"
            quarantine="quarantines/HttpExportService"
            root="roots/HttpExportService"
            sendDigestHeader="no"
            url="http://127.0.0.1:7777"/>
    </Pipeline>
    <Pipeline
        name="DummyExportReceiver"
        root="Collection/dummy/dicom">
        <HttpImportService
            class="org.rsna.ctp.stdstages.HttpImportService"
            logConnections="no"
            name="HttpImportService"
            port="7777"
            quarantine="quarantines/HttpImportService"
            root="roots/HttpImportService"/>
    </Pipeline>
    <Pipeline
        name="DummyManifestReceiver"
        root="Collection/dummy/manifest">
        <HttpImportService
```

```
            class="org.rsna.ctp.stdstages.HttpImportService"
            logConnections="no"
            name="HttpImportService"
            port="7778"
            quarantine="quarantines/HttpImportService"
            root="roots/HttpImportService"/>
    </Pipeline>
</Configuration>
```

The configuration has three main pipelines:

1. CollectionImport receives and stores DICOM files.
2. CollectionAnonymization anonymizes and stores files.
3. CollectionExport transmits files to the principal investigator.

The configuration has two additional pipelines just for testing:

1. DummyExportReceiver receives DICOM files from the CollectionExport pipeline.
2. DummyManifestReceiver receives manifest submissions.

The configuration has three Plugins:

- The TCIAPlugin provides the interface into the pipelines for the wizard.
- The ImportManifestLogPlugin accumulates manifest information for imported files and provides the information for the Lookup Table Template file.
- The ExportManifestLogPlugin accumulates manifest information for anonymized submissions to the principal investigator.

# 3 User Accounts

When CTP starts, the TCIAPlugin changes the password of the **admin** user to **tcia** and grants it these roles:

- The **admin** role grants access to the administrative roles on the server.
- The **TCIA** role grants access to the TCIAServlet.
- The **qadmin** role grants access to the QuarantineServlet.
- The **shutdown** role allows the wizard to shut down CTP.

Additional users can be created through the User Manager on the CTP home page by logging in as a user with the **admin** role.

The wizard can automatically log in the user is as admin/tcia, or it can provide a UI that gets the username and password from the user and then make the call to the LoginServlet. The URL of the LoginServlet is:

**/login/ajax?username=...&password=...**

The login returns either a 200 or a 403 response code.

To provide a logout feature, the wizard can use this URL:

**/login/ajax?logout**

This call always returns 200.

# 4 The TCIAServlet API

This section describes the functions provided to the wizard by the TCIAServlet. All functions are accessed by making an HTTP connection to the CTP server. Unless otherwise indicated, all functions return an HTTP response with **Content-Type: text/xml;charset=UTF-8**.

## 4.1 Submitting Files to the Import Pipeline

### 4.1.1 Listing Files on the Server

To list the files in a directory on the server, the wizard does an HTTP GET to:

**/Collection/listFiles?dir=path**

The servlet returns an XML structure like this:

```
<dir
    name="...directory name..."
    parent="...absolute path to the parent directory...">
    <dir name="...child directory 1 name..."/>
    <dir name="...child directory 2 name..."/>
    ...
    <dir name="...child directory n name..."/>
    <file name="...child file 1 name..."/>
    <file name="...child file 2 name..."/>
    ...
    <file name="...child file n name..."/>
</dir>
```

If the **dcm** query parameter is supplied, `<file>` elements are only returned for files that parse as DICOM files. If the **dcm** query parameter is missing, `<file>` elements are returned for all files.

### 4.1.2 Checking the Space Required for Import

To submit a list of files to the DirectoryImportService in the import pipeline, the wizard does an HTTP GET to:

**/Collection/getSpaceRequired?file="path sequence"**

where **path sequence** is a string of paths separated by pipe characters, like this:

**absolutePath-1|absolutePath-2|...|absolutePath-n**

The servlet computes the sum of the sizes of all the files. An absolutePath that points to a file includes the size of that file. An absolutePath that points to a directory includes the sizes of all the files in the directory and all of its child directories recursively.

This servlet returns an XML structure like this:

```
<space partition="D:\" files="533"
      required="163" available="433385" total="488383"
      units="MB"/>
```

### 4.1.3 Submitting Files to the DirectoryImportService

To submit a list of files to the DirectoryImportService in the import pipeline, the wizard does an HTTP GET to:

**/Collection/submitFile?file="path sequence"**

where **path sequence** is a string of paths separated by pipe characters, like this:

**absolutePath-1|absolutePath-2|...|absolutePath-n**

The servlet copies the files to the import pipeline. An absolutePath that points to a file imports that file. An absolutePath that points to a directory imports all the files in the directory and all of its child directories recursively.

The servlet returns an XML structure indicating whether the submissions all succeeded (<OK/>) or at least one failed (<NOTOK/>).

### 4.1.4 Submitting Files to the DicomImportService

The DicomImportService receives DICOM transfers on port 104. No configuration of the AE Titles is necessary; the SCP accepts all AE Titles.

## 4.2 Viewing the Status of the Import Pipeline

To obtain the number of files that have been received but not yet processed by the import pipeline, the wizard does an HTTP GET to the URL:

**/Collection/getImportStatus**

The AJAX call returns an XML structure like this:

```
<status queueSize="0"/>
```

The queueSize attribute is the sum of the queue sizes of all the import services in the import pipeline. When the queueSize attribute is zero, all the received files have been passed down the pipeline and are ready for anonymization.

## 4.3 The Import Manifest

As files are imported, a manifest listing key identifiers from each file is created. The manifest can be obtained as XML, CSV, or XLSX. The manifest contains one entry for each series. The TCIAServlet provides three functions for accessing the manifest.

### 4.3.1 Listing the Import Manifest as XML

The wizard can obtain an XML structure containing the manifest of series, including PHI, by an HTTP GET to:

**/Collection/listImportManifest/xml**

The resulting structure looks like this:

```
<Manifest>
  <Series>
    <Collection value=""/>
    <SiteName value=""/>
    <PatientID value="1200824338"/>
    <StudyDate value="20010312"/>
    <SeriesInstanceUID value="9999.134743367049617772172711771185484448960"/>
    <StudyDescription value="Head^04_CORONAL_SINUSES (Adult)"/>
    <SeriesDescription value="Sinus Coronal"/>
    <Modality value="CT"/>
    <NumFiles value="38"/>
  </Series>
  <Series>
    <Collection value=""/>
    <SiteName value=""/>
    <PatientID value="1200824338"/>
    <StudyDate value="20010312"/>
    <SeriesInstanceUID value="9999.264302660968596814195213146253418668559"/>
    <StudyDescription value="Head^04_CORONAL_SINUSES (Adult)"/>
    <SeriesDescription value="Topogram 0.6 T20s"/>
    <Modality value="CT"/>
    <NumFiles value="2"/>
  </Series>
</Manifest>
```

### 4.3.2 Listing the Import Manifest as CSV

The wizard can obtain a text string containing the import manifest as a spreadsheet in CSV format by an HTTP GET to:

**/Collection/listImportManifest/csv**

The response returns a CSV text string with **Content-Type: text/csv;charset=UTF-8**. The resulting spreadsheet has one row for each series, as shown in the next section.

### 4.3.3 Listing the Import Manifest as XLSX

The wizard can obtain a text string containing the import manifest as a spreadsheet in XLSX format by an HTTP GET to:

**/Collection/listImportManifest/xlsx**

The response returns a CSV text string with **Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet**. The resulting spreadsheet has one row for each series:

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Collection | SiteName | PatientID | StudyDate | SeriesInstanceUID | StudyDescription | SeriesDescription | Modality | NumFiles |
| 2 | | | | | | | | | |
| 3 | | | 1200824338 | 20010312 | 9999.134743367049617772172711771185484448960 | Head^04_CORONAL_SINUSES (Adult) | Sinus Coronal | CT | 38 |
| 4 | | | 1200824338 | 20010312 | 9999.264302660968596814195213146253418668559 | Head^04_CORONAL_SINUSES (Adult) | Topogram 0.6 T20s | CT | 2 |

## 4.4 Updating the DicomAnonymizer Lookup Table

### 4.4.1 Listing the Lookup Table Template Spreadsheet

To get the spreadsheet containing the PatientIDs of all the imported files, the wizard does an HTTP GET to the URL:

**/Collection/listLookupTableTemplate**

The template spreadsheet looks like this:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | | ptid | cname | ddate |
| 2 | **Local Patient ID** | **Anonymized Patient ID** | **Collection Name** | **Diagnosis Date (M/D/YYYY)** |
| 3 | 1200824338 | | | |

To get the spreadsheet containing the PatientIDs of selected patients, the wizard does an HTTP GET to the URL:

**/Collection/listLookupTableTemplate?id=PatientID1|PatientID2|...|PatientIDn**

### 4.4.2 Submitting the Lookup Table

To update the anonymizer lookup table from an Excel spreadsheet, the wizard does an HTTP POST of the file (with Content-Type **multipart/form-data)** to the URL:

**/Collection**

The spreadsheet must be an XLSX file with this structure:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | | ptid | cname | ddate |
| 2 | **Local Patient ID** | **Anonymized Patient ID** | **Collection Name** | **Diagnosis Date (M/D/YYYY)** |
| 3 | 657662 | 12345 | A Collection Name | 4/15/2014 |
| 4 | 5555 | 4444 | Another Collection Name | 2/1/2011 |
| 5 | etc. | | | |
| 6 | | | | |

The first row contains the KeyTypes used in the lookup table for the data in each spreadsheet column. This row must not be modified.

The second row provides a human-readable label for the column. The TCIA servlet needs the first row. It ignores the second row and starts processing rows at the third row.

## 4.5 Viewing the List of Files Ready for Anonymization

To obtain the list of files that have been received but not yet anonymized, the wizard does an HTTP GET to the URL:

**/Collection/listImport**

The response returns an XML structure like this:

```
<DicomFiles count="40">
  <dir count="40" name="DirectoryStorageService" parent="D:\JavaPrograms\CTP-TCIA\CTP\Collection\import\roots">
    <dir count="40" name="1200824338-Bunny,Bugs">
      <dir count="40" name="20010312">
        <dir count="2" name="Series1">
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="1" StudyDate="20010312" name="9999.222149640329028447124359750781404748588"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="1" StudyDate="20010312" name="9999.803493271008570235407518041026327035 89"/>
        </dir>
        <dir count="38" name="Series2">
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.104115109181271829052136463305872175892"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.113853809324458845023883428390519345 79"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.113983435978265066276300216885396006062"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.129003575003187817511384952750931955308"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.133645333193764450839499526458610875307"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.134095359512589080454306553825903370604"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.140285531251106151469597569011030160535"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.145415533035164672938705109621299252634"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.147559515087250199331639124618594546423"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.156933851023183000154402337165409857 31"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.162760251557332918073608336500345336242"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.165810487962087107324577666065514793531"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.169831188606048016196033942139497761570"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.191458548338121923480652577383665455 58"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.194510777975218778730611975661279731 79"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.199259582581030369204776800763903847687"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.208689751371375329075102446940617839366"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.214257504674206793970526899765796655764"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.220943752888531707761401256395596647078"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.246112055202027104929794630877353288271"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.246505612395667702453661050403128845256"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.259254080765291461288576563062035589231"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.267040403754384675701867718924305846350"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.273080104198635637751548367339060238477"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.273252232837285712874226462087896306 3"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.283147573022529976778776778811457402258"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.289552233369947118261678031165462679034"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.289851792726025215340234985487577939151"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.292524086301371757006501187334377036996"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.303802472840681483515970797206942914 13"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.364847711564780298663357628982875230 06"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.538816343721148695227533339349426833 89"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.722560140209572794624489654926597399 2"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.738273761234903864490675407382205345 66"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.739845529060822567735127795543319526 11"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.789973425611418385399727521124813262 38"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.884182050252888927844033847552263034 26"/>
          <DicomObject Modality="CT" PatientID="1200824338" PatientName="Bunny,Bugs" Series="2" StudyDate="20010312" name="9999.951928045060730956281380824222296874 92"/>
        </dir>
      </dir>
    </dir>
  </dir>
</DicomFiles>
```


The XML structure mimics the directory structure in which the files are stored. Each **dir** element in the XML structure represents a directory. The top-level directory is the root directory of the DirectoryStorageService pipeline stage that contains the files received by the import pipeline. Under the top-level directory, there is one directory for each patient. Under a patient's directory is one directory for each date on which the patient had a study. The study date directory contains all the images for studies done on that date. Individual image files are represented in the XML structure by **DicomObject** elements.

The relative path to a directory can be obtained by walking the tree from the point in question back to the top. Thus, the relative path to the **20010312** directory is:

**DirectoryStorageService/1200824338-Bunny,Bugs/20010312**

The base of the relative path is the root of the pipeline stage.

## 4.6 Initializing the Anonymization Pipeline Counts

Before files are submitted to the anonymizer pipeline, the pipeline must be initialized by an HTTP GET to the URL:

**/Collection/initializeAnonymizerPipelineCounts**

The servlet returns an XML structure like this:

```
<Status
    startingQuarantineCount="0"
    currentQuarantineCount="0"
    queuedInstanceCount="40"
    currentManifestInstanceCount="40"/>
```

## 4.7 Submitting Files for Anonymization

To anonymize studies that were listed in 4.2, above, the wizard passes some level in the hierarchy (a patient, a single study, or the top-level directory) to the servlet in an HTTP GET to the URL:

**/Collection/anonymize?file=filepath**

where **filepath** is the relative path. If the filepath is a directory, the servlet processes all the files in that directory and all its child directories. For example, to process everything that has been received, the URL would be:

**/Collection/anonymize?file=DirectoryStorageService**

The servlet moves the files from the import pipeline to the anonymizer pipeline to start the anonymization. The servlet returns an XML structure with one element <OK/> or <NOTOK/> to indicate whether all the moves were successful.

## 4.8 Pausing the Anonymization Pipeline

To pause the anonymization pipeline, the wizard does an HTTP GET to the URL:

**/Collection/pause**

The servlet returns an XML structure with one element <OK/>. While the pipeline is paused, no files flow down the pipe, but files are accepted into the import service of the pipe, and any files that were being exported from the pipe continue to be exported.

## 4.9 Restarting a Paused Anonymization Pipeline

To restart the anonymization pipeline if it has been paused, the wizard does an HTTP GET to the URL:

**/Collection/restart**

The servlet only restarts the pipeline if it had previously been paused; otherwise, it does nothing. The servlet returns an XML structure with one element <OK/> if the pipeline was actually restarted; otherwise, it returns <NOTOK/>.

If the path element **/requeue** is added, the contents of the anonymizer quarantine are requeued in the import service of the anonymizer pipeline before the restart:

**/Collection/restart/requeue**

## 4.10 Viewing the List of Files Ready for Export

To obtain the list of files that have been anonymized but not yet exported, the wizard does an HTTP GET to the URL:

**/Collection/listAnonymized**

The response returns an XML structure like this:

```
<DicomFiles count="40">
  <dir count="40" name="DirectoryStorageService" parent="D:\JavaPrograms\CTP-TCIA\CTP\Collection\anonymization\roots">
    <dir count="40" name="1200824338-Bunny,Bugs">
      <dir count="40" name="20010312">
        <dir count="2" name="Series1">
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="1" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.196154021349712661720723242598"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="1" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.249876087015328106310133805613"/>
        </dir>
        <dir count="38" name="Series2">
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.100668460971350414894490102932"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.102060935583543320799436830948"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.105779589864872573224308501378"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.115459220012628587817540242011"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.117175100163883877277276475378"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.123921539284844847827383056846"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.128675395687132861321061084828"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.130421976200204254919547971728"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.139441815545700888557228572515"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.142588588142605889042532570140"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.149949289905196243172970066630"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.159422924702382504851306269862"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.161158294508295584409728267395"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.161699679064174416207625600477"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.168239976109073543229622188414"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.187189977615953577403173506995"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.198466521145949332279162363066"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.207220241178249664138862977244"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.246463463499649159607524292287"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.246588027658302458842157495980"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.259485621065070668421580885329"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.262577426611873494116410095576"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.267271874089169982177365211904"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.267680436852985977609972671988"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.269659613708517497492872534597"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.277308539347417393830814783342"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.305043950418409172941668374556"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.313698179949817552877724376832"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.326275684914431774802750684407"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.326703538879940132708569163446"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.600969418674983315880418546034"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.616743252698900306645555850882"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.667619360275242009041302999923"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.688292042196366947154103689180"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.694540314427075073591842689384"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.696871331479752617938799266983"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.774554274778940355833861929180"/>
          <DicomObject Modality="CT" PatientID="987654" PatientName="987654" Series="2" StudyDate="19290209" name="1.3.6.1.4.1.14519.5.2.1.9999.9999.992124065854561686185831304265"/>
        </dir>
      </dir>
    </dir>
  </dir>
</DicomFiles>
```


The XML structure exactly mimics the one that listed the imported files, but it references a different DirectoryStorageService pipeline stage, so although the directory names are the same, they are relative to a different root directory.

[This is probably also a good place to provide file viewing capabilities to reassure the user that they are not exporting PHI.]

## 4.11 Exporting Files

To export studies that were listed in 4.4, above, the wizard passes some level in the hierarchy (a patient, a single study, or the top-level directory) to the servlet in an HTTP GET to the URL:

**/Collection/export?file=filepath**

where **filepath** is again the relative path.

The servlet moves the files from the anonymizer pipeline to the export pipeline to start the export. The servlet returns an XML structure with one element <OK/> or <NOTOK/> to indicate whether all the moves were successful.

## 4.12 The Export Manifest Database

As files are anonymized, a manifest database listing key identifiers from each file is created. The manifest database can be viewed as XML, CSV, or XSLX, with or without PHI. The manifest with PHI is called the Local Manifest; the manifest without PHI is called the Export Manifest, since it is the version that is exported to the principal investigator. The manifest contains one entry for each series. The TCIAServlet provides eight functions for accessing the manifest:

### 4.12.1 Clearing the Export Manifest Database

The wizard can clear the manifest by an HTTP GET to the URL:

**/Collection/clearExportManifest**

The servlet returns an XML structure with one element <OK/> or <NOTOK/> to indicate whether all the operation was successful. After calling this function, the manifest database is empty, and the **initializeAnonymizerPipeline** should be called if more files are to be processed.

### 4.12.2 Getting the Export Manifest Database Status

The wizard can get an XML structure containing the status of the processing of files in the Anonymizer pipeline by an HTTP GET to the URL:

**/Collection/getExportManifestStatus**

The servlet returns an XML structure like this:

```
<Status
    startingQuarantineCount="0"
    currentQuarantineCount="0"
    queuedInstanceCount="40"
    currentManifestInstanceCount="40"/>
```

### 4.12.3 The Local Manifest

#### *4.12.3.1 Listing the Manifest with PHI as XML*

The wizard can obtain an XML structure containing the manifest of series, including PHI, by an HTTP GET to:

**/Collection/listLocalManifest/xml**

The resulting structure looks like this:


```
▼<Manifest>
  ▼<Series>
     <Collection value="C1"/>
     <SiteName value="TCIA"/>
     <PatientID phi="1200824338" value="987654"/>
     <StudyDate phi="20010312" value="19290209"/>
     <SeriesInstanceUID phi="9999.134743367049617772172717718548448960" value="1.3.6.1.4.1.14519.5.2.1.9999.9999.113479017276406998463382173796"/>
     <StudyDescription value="Head^04_CORONAL_SINUSES (Adult)"/>
     <SeriesDescription value="Sinus Coronal"/>
     <Modality value="CT"/>
     <NumFiles value="38"/>
   </Series>
  ▼<Series>
     <Collection value="C1"/>
     <SiteName value="TCIA"/>
     <PatientID phi="1200824338" value="987654"/>
     <StudyDate phi="20010312" value="19290209"/>
     <SeriesInstanceUID phi="9999.264302660968596814195213146253418668559" value="1.3.6.1.4.1.14519.5.2.1.9999.9999.116933866301694768395323515093"/>
     <StudyDescription value="Head^04_CORONAL_SINUSES (Adult)"/>
     <SeriesDescription value="Topogram 0.6 T20s"/>
     <Modality value="CT"/>
     <NumFiles value="2"/>
   </Series>
 </Manifest>
```


#### *4.12.3.2 Listing the Manifest with PHI as CSV*

The wizard can obtain a text string containing the manifest as a spreadsheet in CSV format by an HTTP GET to:

**/Collection/listLocalManifest/csv**

The response returns a CSV text string with **Content-Type: text/csv;charset=UTF-8**. The resulting spreadsheet has one row for each series.

#### *4.12.3.3 Listing the Manifest with PHI as XLSX*

The wizard can obtain a text string containing the manifest as a spreadsheet in XLSX format by an HTTP GET to:

**/Collection/listLocalManifest/xlsx**

The response returns an XLSX file with **Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet**. The resulting spreadsheet has one row for each series.

### 4.12.4 The Export Manifest

#### *4.12.4.1 Listing the Manifest without PHI as XML*

The wizard can obtain an XML structure containing the manifest of series, including PHI, by an HTTP GET to:

**/Collection/listExportManifest/xml**

The resulting structure looks like this note that the phi attributes of the elements are not present):

```
▼<Manifest>
  ▼<Series>
     <Collection value="C1"/>
     <SiteName value="TCIA"/>
     <PatientID value="987654"/>
     <StudyDate value="19290209"/>
     <SeriesInstanceUID value="1.3.6.1.4.1.14519.5.2.1.9999.9999.113479017276406998463382173796"/>
     <StudyDescription value="Head^04_CORONAL_SINUSES (Adult)"/>
     <SeriesDescription value="Sinus Coronal"/>
     <Modality value="CT"/>
     <NumFiles value="38"/>
   </Series>
  ▼<Series>
     <Collection value="C1"/>
     <SiteName value="TCIA"/>
     <PatientID value="987654"/>
     <StudyDate value="19290209"/>
     <SeriesInstanceUID value="1.3.6.1.4.1.14519.5.2.1.9999.9999.116933866301694768395323515093"/>
     <StudyDescription value="Head^04_CORONAL_SINUSES (Adult)"/>
     <SeriesDescription value="Topogram 0.6 T20s"/>
     <Modality value="CT"/>
     <NumFiles value="2"/>
   </Series>
 </Manifest>
```

#### *4.12.4.2 Listing the Manifest without PHI as CSV*

The wizard can obtain a text string containing the manifest as a spreadsheet in CSV format by an HTTP GET to:

**/Collection/listExportManifest/csv**

The response returns a CSV text string with **Content-Type: text/csv;charset=UTF-8**. The resulting spreadsheet contains one row for each series, with only the anonymized values.

#### *4.12.4.3 Listing the Manifest without PHI as XLSX*

The wizard can obtain a text string containing the manifest as a spreadsheet in XLSX format by an HTTP GET to:

**/Collection/listExportManifest/xlsx**

The response returns a CSV text response with **Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet**. The resulting spreadsheet contains one row for each series, with only the anonymized values.

### 4.12.5 Exporting the Manifest

The wizard can sent the CSV manifest without PHI to the principal investigator by an HTTP GET to:

**/Collection/exportManifest**

The servlet returns an XML structure with one element <OK/> or <NOTOK/> to indicate whether the transmission was successful.

## 4.13 Image Functions

### 4.13.1 Listing the Elements of a DICOM File

The wizard can obtain an HTML page containing a table showing the values of the elements in a DICOM file by an HTTP GET to the URL:

**/Collection/listElements?file=filepath**

The servlet returns a JPEG image in the response with **Content-Type: image/jpeg**. If the servlet cannot find the file, the servlet returns a 404 response code. If the servlet cannot parse the file, the servlet returns a 500 response code and enters a stack trace in the log.

### 4.13.2 Getting a JPEG Image from a DICOM File

The wizard can obtain a browser-viewable image of a DICOM by an HTTP GET to the URL:

**/Collection/getImage?file=filepath**

The servlet returns a JPEG image in the response with **Content-Type: image/jpeg**. If the servlet cannot find the file, the servlet returns a 404 response code. If the servlet cannot parse the file, the servlet returns a 500 response code and enters a stack trace in the log. If a 500 is received and the log shows a **No class def found** error for the StreamSegmentMapper, it means that either the ImageIO Tools have not been installed, or the image has a format is not supported by the ImageIO Tools.

## 4.14 Special Functions

### 4.14.1 Getting the List of File System Roots on the Server

The wizard can obtain an XML structure listing the file system roots on the server by an HTTP GET to the URL:

**/Collection/getFileSystemRoots**

This servlet returns an XML structure like this:

```
<roots>
  <root name="C:\" desc="Local Disk"/>
  <root name="D:\" desc="Local Disk"/>
  <root name="E:\" desc="Local Disk"/>
  <root name="P:\" desc="CD Drive"/>
  <root name="Q:\" desc="CD Drive"/>
</roots>
```

### 4.14.2 Getting the Available Space on the Server

The wizard can obtain an XML structure indicating the available space on the partition on which the CTP instance is located by an HTTP GET to the URL:

**/Collection/getAvailableSpace**

This servlet returns an XML structure like this:

```
<space partition="D:\" available="434932" units="MB"/>
```

The wizard can specify the file system root on which to obtain the available space by including the **root** query parameter:

**/Collection/getAvailableSpace?root=...**

Where the value of the **root** parameter is a name obtained from the getFileSystemRoots function.

### 4.14.3 Getting the Export Queue Size

The wizard can obtain an XML structure indicating the current size of the export queue by an HTTP GET to the URL:

**/Collection/getExportQueueSize**

This servlet returns an XML structure like this:

```
<queue stage="HttpExportService" size="10"/>
```

### 4.14.4 Getting the URL of the Quarantine Manager Servlet

The wizard can obtain an XML structure containing the URL of the CTP Quarantine Manager for the DicomAnonymizer quarantine by an HTTP GET to the URL:

**/Collection/getQuarantineURL**

This servlet returns an XML structure like this:

```
<quarantine stage="DicomAnonymizer" url="/quarantines?p=1&s=2"/>
```

### 4.14.5 Getting a Summary of the DicomAnonymizer Quarantine

The wizard can obtain an XML structure containing a summary of the objects in the DicomAnonymizer quarantine by an HTTP GET to the URL:

**/Collection/getQuarantineSummary**

This call returns an XML structure like this:

… TBD …

### 4.14.6 Shutting Down the Server

The wizard can shut down the CTP server by an HTTP GET to:

**/shutdown**

The system returns either a 200 or 403 response code. It fails if the user does not have the **shutdown** role. This URL accesses the CTP ShutdownServlet directly. It returns a web page like this:

> Shutdown request received from admin at 192.168.0.225.
> Goodbye.

## 4.15 Testing Functions

During testing, it may be convenient to clear all the import/export directories, the DicomAnonymizer quarantine, and the manifests. This can be done by an HTTP GET to the URL:

**/Collection/reset**

The servlet does all it can and returns an XML structure with one element <OK/>. It never returns <NOTOK/>.