NBIA Feature DB Adapter Description

|  |  |
| --- | --- |
| Last Revised: | Sep 29, 2009 |
| Produced By: | NBIA Team |
| Version: |  |

Document Approvals

The list contains the name and contact information for the core project team and any key stakeholders who have an interest in the success of the project. An “S” identifies persons responsible for approval from the stakeholder groups. Sign off of the document would be required when a decision is made not to take action for defined gaps.

|  |  |  |
| --- | --- | --- |
| S | Name | Role |
| S | Robert Shirley | NCICBIIT |
| S | Peter Yan | COTR/PM |
|  | Eric Kascic | Technical Lead |
|  |  |  |

Revision History

When you make a change to a document, you must add an entry to this Revision History table and you must manually type the Last Revised Date on the front cover.

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Revised by |
| 12/08/2010 | 1 | Draft Document for DB Adapter feature | Jim Zhou |
|  |  |  |  |
|  |  |  |  |

Copyrights and Trademarks

© Copyright 2010 by CBIIT, caBIGTM. All rights reserved.

Table of Contents

[1. Introduction 1](#_Toc279500559)

[2. Summary of Module 1](#_Toc279500560)

[3. Architectural Diagram 1](#_Toc279500561)

[4. Example 1 1](#_Toc279500562)

[5. Example 2 1](#_Toc279500563)

# Introduction

The DatabaseAdapter class is for storing DICOM header information into NBIA database. It interacts with database export service of CTP. The function for DatabaseAdapter class is to extract NBIA needed information from a DICOM object, validate them and store them into NBIA database for future searching.

The DatabaseAdapter class provides a set of methods allowing the DatabaseExportService to perform various functions, all of which are explained in the Javadocs. The basic interaction model is:

* When the DatabaseExportService detects that files are in its queue, it determines whether the database interface class is loaded and loads it if necessary.
* It then calls the database interface’s **connect()** method.
* For each file in the queue, it instantiates an object matching the file’s contents and calls the database interface’s **process()** method. There are four overloaded process methods, one for each object class.
* When the queue is empty, it calls the database interface’s **disconnect()** method.

All the methods of the DatabaseAdapter class return a static instance of the **org.rsna.ctp.pipeline.Status** class to indicate the result. The values are:

* **Status.OK** means that the operation succeeded completely.
* **Status.FAIL** means that the operation failed and trying again will also fail. This status value indicates a problem with the object being processed.
* **Status.RETRY** means that the operation failed but trying again later may succeed. This status value indicates a temporary problem accessing the external database.

# Summary of Feature

NBIA database adapter class extends from org.rsna.ctp.stdstages.database.DatabaseAdapter class. There are four functions that are inherits from DatabaseAdapter class.

* Process DICOM object
* Process XML Object (Annotation)
* Process Zip Object
* Process File object.

NBIA DB adapter has following features,

* Validate patient information from DICOM file
* Validate Study information from DICOM file
* Validate Series information from DICOM file
* Validate image information from DICOM file
* Validate CT image information from DICOM file
* Validate Annotation information from DICOM file
* Validate Equipment information from DICOM file
* Validate Trial Data Provenance information from DICOM file
* Store/update patient information
* Store/update study information
* Store/update series information
* Store/update image information
* Store/update CT image information
* Store/update Annotation information
* Store/update equipment information
* Store/update Trial Data Provenance information
* Store Annotation submission history data
* Store image submission history data

NBIA adapter class is independent from NBIA application code. It integrates in RSNA CTP code. So the hibernate mapping data are slightly different from NBIA domain object due to write access.

# Architectural Diagram



**Figure 3.1 DB Adapter Use Case**

## Workflow

When CTP sends a request to NBIA DB Adapter, NCIADatabase service (DB adapter) will invoke NCIADatabaseDelegator service. The delegator will pick the DICOM object that CTP parsed. If it is a DICOM file, the imageStorage service will be invoked, or if it is an annotation file, AnnotationStorage service will be invoked. The third case is after the submitter submits files; they can use CTP to check the status of submission.

* DICOM file submission

When a DICOM file goes through DB adapter, DB adapter will store information of patient, the equipment image is taken with, Trail Data Provenance, study, series, image, and CT image into NBIA database with above order.

* Annotation Submission

Annotation information will be inserted into database with this service. Annotation file itself is in XML format.

* Submission status check

The UIDQuery will be invoked in CTP when the client query the status of submission, and it will also invoke DB adapter’s DicomSOPInstanceUIDQuery service to check NBIA database with the specified SOP instance UID.

## Validation

When DB adapter stores DICOM information into database, the DB adapter will validate most of information for patient, study, series, image, and trial data provenance.

In order to reduce the duplicate records, validate processing also check if record exists or not, if yes, the update process will be performed. For instance, check patient with patient ID, validate process will check database if this patient ID exists in patient table, validator will fetch record out, and DB adapter will update information and store it back to the NBIA database. If no patient ID found, a new record will be inserted into database.

## Quarantining files

What if there are some bad files that are submitted by submitters? For CTP, there are layers for quarantining process. One is from CTP, another one is from DB adapter. CTP will check if the DICOM if is corrupted, or DICOM data is completed missing. DB adapter will check data what NBIA application concerns.

Main reasons for quarantining files,

* any of following items missing, the file will be quarantined.
  + Patinent ID
  + Study Instance UID
  + Series Instance UID
  + SOP Instance UID
  + modality
* Group 13 contains no data,
  + Project
  + Site ID
  + Site Name
* Others
  + Duplicated records already exist
  + When updating, the updated records are in to be deleted, or delete status.

## Class Diagram

## 

**Figure 3.2 DB Adapter Class Diagram**

# Change Example 1

## Set Eclipse Environment

* Check out NBIA source from trunk by using Tortoise tool
* Change directory to SOURCE\_CODE\_DIR/build
* Modify install.properties for database connection (user id, password, etc)
* Type “ant build:all” to build project (This must be happened before importing source code into Eclipse)
* Open Eclipse to import all modules as existing project
* Fix build path issue in the Eclipse, such as NBIA\_BASE set up
* After this, you can utilize the Eclipse to change your code.

## Steps regenerate CTP client and server code / Run Test Case

After modifying your CTP code, you need to perform the followings steps to regenerate CTP client and server

* From DOS prompt, run ant build:ncia-CTP (you can run ant build:all)
* Run ant build deploy:local:install
* Above process will generate a directory C:\apps by default
* Change directory to c:\apps\ncia, you will see two directories, CTP-Client, CTP-Server
* Start CTP server first by running ctp.bat under CTP-Server/CTP directory
* Start CTP client by running ctp.bat under CTP-Client/CTP directory
* Start File Sender by running FileSender.bat under CTP-Client/FileSender directory.

In this point, you can test your NBIA data submission.

# Change Example 2

No further example.