FDA Web Content Mining

BITS ZG628T: Dissertation

by

T.SRI KUMARAN

2013HT12504

Dissertation work carried out at

HCL Technologies Ltd., Chennai

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI (RAJASTHAN)

April 2015

FDA Web Content Mining

BITS ZG628T: Dissertation

by

T.SRI KUMARAN

2013HT12504

Dissertation work carried out at

HCL Technologies Ltd., Chennai

Submitted in partial fulfillment of M.Tech. Software Systems degree programme

Under the Supervision of
I.SATHISH KUMAR, ASSOCIATE GENERAL MANAGER
HCL Technologies Ltd., Chennai

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE
PILANI (RAJASTHAN)

April, 2015

Certificate

This is to certify that the Dissertation entitled FDA Web Content Mining and submitted by T.SRI KUMARAN having ID-No. 2013HT12504 for the partial fulfillment of the requirements of M.Tech. Software Systems degree of BITS, embodies the bonafide work done by him under my supervision.

[illegible]

Signature of the Supervisor

Place: Chennai

Date: March, 2015

I.Sathish Kumar,
Associate General Manager,
HCL Technologies Ltd.,
Chennai 6000119.

Birla Institute of Technology & Science, Pilani

Work-Integrated Learning Programmes Division

Second Semester 2014-2015

BITS ZG628T: Dissertation

Abstract

BITS ID No.	: 2013HT12504
NAME OF THE STUDENT	: T.SRI KUMARAN
EMAIL ADDRESS	: srikumaran.t@gmail.com
STUDENT'S EMPLOYING ORGANIZATION & LOCATION	: HCL Technologies Ltd., Chennai.
SUPERVISOR'S NAME	: I.SATHISH KUMAR
SUPERVISOR'S EMPLOYING ORGANIZATION & LOCATION	: HCL Technologies Ltd., Chennai
SUPERVISOR'S EMAIL ADDRESS:	sathishi@yahoo.com
DISSERTATION TITLE	: FDA Web Content Mining

Abstract :

The Food and Drug Administration (http://www.fda.gov/) is a federal agency of the United States Department of Health and Human Services.

FDA Web Site has many subpages for each sector of Food and Drug. FDA is the only place where Medical Writers, Directors and Executives of Pharmaceutical Industries can get complete guidelines of Drug Submission and related details. They should manually check this Web Site on a daily basis to gain updates on news, events and especially on new guidelines for Drug submission posted by International Conference on Harmonisation (ICH).

Presently new changes imparted in the guidelines were extracted from this site and imported or versioned to Documentum Content Management System (CMS) manually by Medical Writers whenever an update is released.

The contents were published at FDA Website using Documentum Web Publisher by FDA.
To check for News, navigate to all the subpages and extract content from FDA Web Site manually is a challenging and time consuming routine.

We will research on how to build an efficient, flexible and extensible technical solution to automate FDA Web Content Extraction, Transformation and Loading.

Broad Academic Area of Work: Data Warehousing and/or Data Mining

Key words: FDA, Web Content Mining, Web Content Management, Web Content Warehousing, ETL, Extraction, Transformation, Loading

[illegible]	[illegible]
Signature of the Student	**Signature of the Supervisor**
Name: Sri kumaran . T	**Name:** I - SATHISH KUMAR
Date: 09 - Jan - 2015	**Date:** 09 - JAN - 2015
Place: chennai	**Place:** CHENNAI

Acknowledgements

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without mentioning the people who made it possible, because success is the epitome of hard work, perseverance, undeterred missionary zeal, steadfast determination and most of all “ENCOURAGING GUIDANCE”.

I express my gratitude to my supervisor Mr. Sathish Kumar Inkershal for providing me a means of attaining my most cherished goals.

I record my heart full of thanks and gratitude to my additional examiner Mr. Sathish Kumar Rajan and BITS WILP for providing me an opportunity to carry this project, along with purposeful guidance and moral support extended to me throughout the duration of the project work.

I would like to express my love and gratitude to my beloved family, my friends, my HCL team members, for their understanding & motivation throughout the duration of this project.

SRI KUMARAN THIRUPPATHY

List of Abbreviations and Acronyms

CFR	Code of Federal Regulation
CMS	Content Management System
CSV	Computer System Validation
DFC	Documentum Foundation Classes
DOM	Document Object Model
DQL	Documentum Query Language
ETL	Extraction, Transformation, Loading
FDA	Food and Drug Administration
GAMP	Good Automated Manufacturing Processes Forum
GCP	Good Clinical Practices
GLC	Good Laboratory Practices
GMP	Good Manufacturing Practices
ICH	International Conference on Harmonisation
IE	Internet Explorer
IQ	Installation Qualification
JCF	Java Collection Framework
JDK	Java Development Kit
OQ	Operational Qualification
PQ	Performance Qualification
SDLC	Software Development Life Cycle
URL	Uniform Resource Locator
WDK	Webtop Development Kit

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background

The Food and Drug Administration (FDA or USFDA *http://www.fda.gov/)* is a federal agency of the United States Department of Health and Human Services, one of the United States federal executive departments. The FDA is responsible for protecting and promoting public health through the regulation and supervision of food safety, tobacco products, dietary supplements, prescription and over-the-counter pharmaceutical drugs (medications), vaccines, biopharmaceuticals, blood transfusions, medical devices, electromagnetic radiation emitting devices (ERED), cosmetics, animal foods & feed and veterinary products.

FDA Web Site has many subpages for each sector of Food and Drug. FDA is the only place where Medical Writers, Directors and Executives of Pharmaceutical Industries can get complete guidelines of Drug Submissions. They should check this Web Site on a daily basis to gain updates on News, events and especially on new guidelines for the Drug submissions posted by International Conference on Harmonisation (ICH).

New changes imparted in the guidelines will be extracted from this Site and imported or versioned to Content Management Systems (CMS) manually whenever an update is released.

1.2 Objectives

To provide an efficient, flexible and extensible technical solution to automate FDA Web Content Extraction, Transformation and Loading process. This application can be compatible with current and future Content Management Systems for storing mined content from FDA Site.

1.3 Scope of Work

Scope of Work comprises of developing a framework for Web Content Mining specific to FDA Site standards and implementing it with current Content Management System.

Proof – Of – Concept (POC) for this study will be developed according to the Plan of Work. It is actually expected to be enhanced and implemented in customer’s server based on this POC.

Two Major Modules:
1. Web Mining from FDA Site
2. Loading and Versioning of extracted content to Content Management System.

Development Life Cycle: AGILE Methodology.

Project will be carried out in five phases: Analysis, Designing, Coding, Testing and Implementation.

1.4 Plan of Work

Module 1	
Analysis	Analyze the FDA Site standards and match it with project requirements. Analyze Content Management System for classifying and loading extracted content.
Designing	Design a Frame work for Extraction, Transformation and Loading of FDA content to Content Management System.
Coding	1. Selenium, Web drivers, HTTP, required packages from Java, IE 2. Ecllipse.

Testing	Module-1 should be tested thoroughly.
Module 2	
Coding	1. Required packages from Java, 2. Documentum Foundation Classes (DFC), 3. Documentum Query Language (DQL), 4. Ecllipse.
Validation	Module-2 should be validated thoroughly.
Module 3	
This module includes Coding for Logger and few common functionalities.	
Implementation	Final Implementation of POC will be in Test Server.

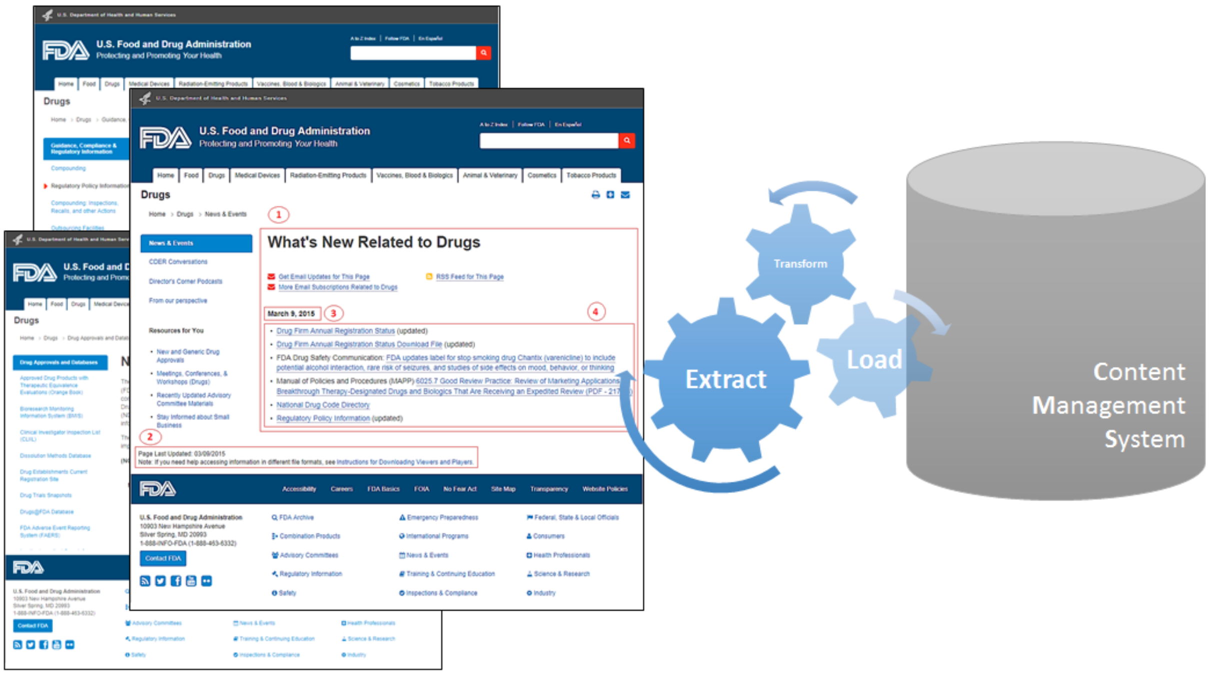

U.S. Food and Drug Administration
Protecting and Promoting Your Health
Drugs
What's New Related to Drugs
Transform
Extract
Load
Content Management System

Figure 1-1: FDA Web Content Mining

Chapter 2: Extraction and Transformation (ET)

2.1 Overview

FDA is U.S. Department of Health and Human Services, mainly deals with Foods and Drugs. In FDA Web Content Mining project scope, ETL tool deals only with Drugs News & Events page.

Let us move on to technical terms. In Extraction and Transformation phase, ETL tool extracts 1. List of Today's News and Events, 2. List of News and Events in it subpages, 3. News details and it content. Basically new updates, content and it metadata. *For Example: List, Href from URL of PDF, ZIP and it text.* Following to extraction phase, List and Metadata will be finely transformed to match with Content Management System standards. *For Example: Removal of unwanted text from document name/title.*

2.2 Why Selenium?

There are thousands of technologies in the world. We have requirement in our hand, now it's time to choose a technology which right opt to our business requirement and within the scope & existing setup at client location.

In our project scope, 100% extraction of contents and it details from Web Page, i.e.: Web Browsers.

Selenium is one of the most powerful web automation packages in the IT market, from Apache. It is an open source package available in Java, C#, Ruby, etc.

Tops of Selenium when compare to Quick Test Professional:

- Supports more languages
- Supports all browsers
- Supports all environments
- Supports most of the IDEs and frameworks
- Outstanding object recognition, *for example: XPath ID, DOM*
- Zero software cost (Open Source!)
- Consumes low hardware resources
- Extensible to mobile implementation
- Dedicated package for web browser automation

In simple, Selenium automates browsers. That's it!

2.3 Selenium Web Drivers and XPath

Selenium WebDriver is the successor to Selenium Remote-Control. Selenium WebDriver accepts commands (sent in Selenese, or via a Client API) and sends them to a browser. This is implemented through a browser-specific browser driver, which sends commands to a browser, and retrieves results. Most browser drivers actually launch and access a browser application (such as Firefox or Internet Explorer); there is also an HtmlUnit browser driver, which simulates a browser using HtmlUnit. The WebDriver directly starts a browser instance and controls it. The WebDriver is fully implemented and supported in Python, Ruby, Java, and C#.

XPath is a really good way to navigate site when there are no IDs on elements that need to work with or is near the element want to work with. *For Example: Locate element <div class="col-md-9 col-md-push-3 middle-column"> by it class name. XPath: //div[contains(@class,'middle-column')]//p.*

2.4 Selenium in our Project Scope

Why Java out of all?

There are more than 1000 comparisons and benefits behind a technology, say Java. The ultimate baseline and reason for choosing Java technology for FDA Web Content Mining is 1. Powerful open source package, 2. Content Management System's technology is also Java. Developing all the modules of ETL in same technology must help for good interoperability between the modules and reduce the usage of transformable components and it complexity. So this reduces overall cost.

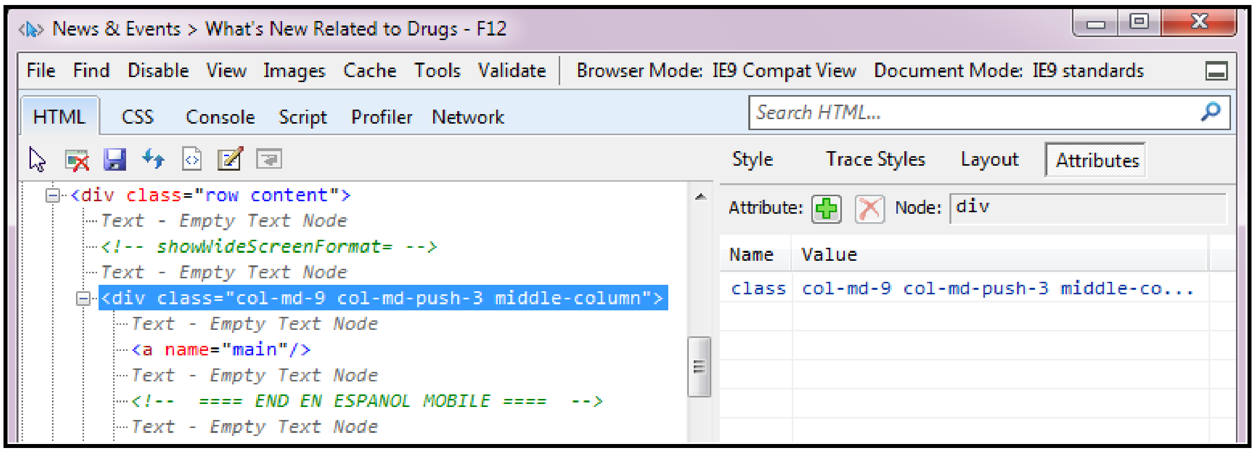
Why Windows? Why IEDriverServer?

ET Modules will be deployed in an existing Windows Server. In future the inline batch execution will be converted as a windows service. This service will be linked to Microsoft Server Manager to monitor it.
IE is a good browser and default licensed version of windows, has built-in developer tool. IEDriverServer is one of the browser specific drivers of WebDrivers.
Mainly, FDA recommended to use FDA Site at IE.
L Module, basically it deals with Content Management System to load extracted data. This will be deployed in an existing Linux Server.

Selenium and XPath

Locating Techniques: FDA Site use dynamic values for element's id attributes, which is difficult to locate. One simple solution is to use XPath functions and base the location on any attribute of the elements.
For example:

1. *Starts-with: XPath: //input[starts-with(@id, 'text-')]*
2. *Contains: XPath: //span[contains(@class, 'heading')]*
3. *Siblings: Forms and Tables*
4. *CSS Locator: CSS: css=div.article-heading*


News & Events > What's New Related to Drugs - F12
File Find Disable View Images Cache Tools Validate
Browser Mode: IE9 Compat View Document Mode: IE9 standards
HTML CSS Console Script Profiler Network
Search HTML...
Style Trace Styles Layout Attributes
Attribute: Node: div

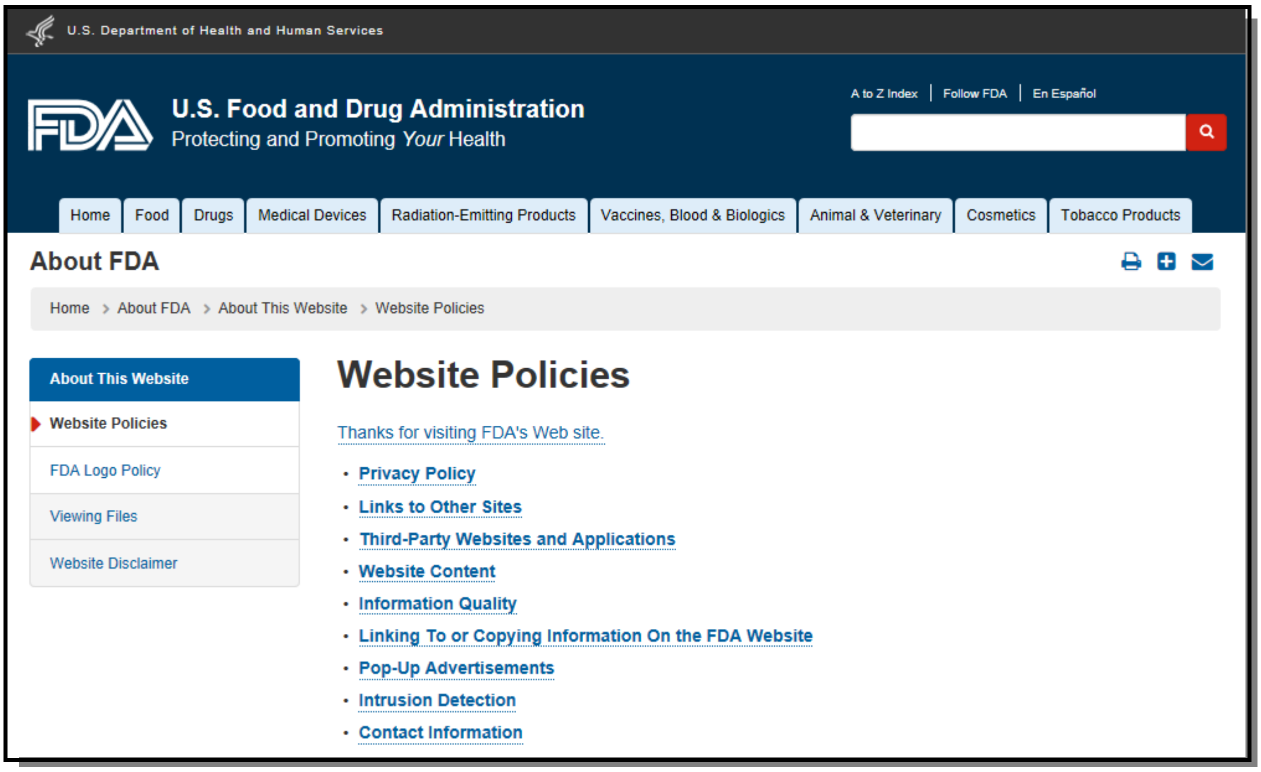
```
<div class="row content">
    Text - Empty Text Node
    <!-- showWideScreenFormat= -->
    Text - Empty Text Node
    <div class="col-md-9 col-md-push-3 middle-column">
        Text - Empty Text Node
        <a name="main"/>
        Text - Empty Text Node
        <!--  ==== END EN ESPANOL MOBILE ====  -->
        Text - Empty Text Node
```

Name	Value
class	col-md-9 col-md-push-3 middle-co...

Figure 2-1: Locating Techniques - XPath

2.5 Analysis on FDA News & Events Page

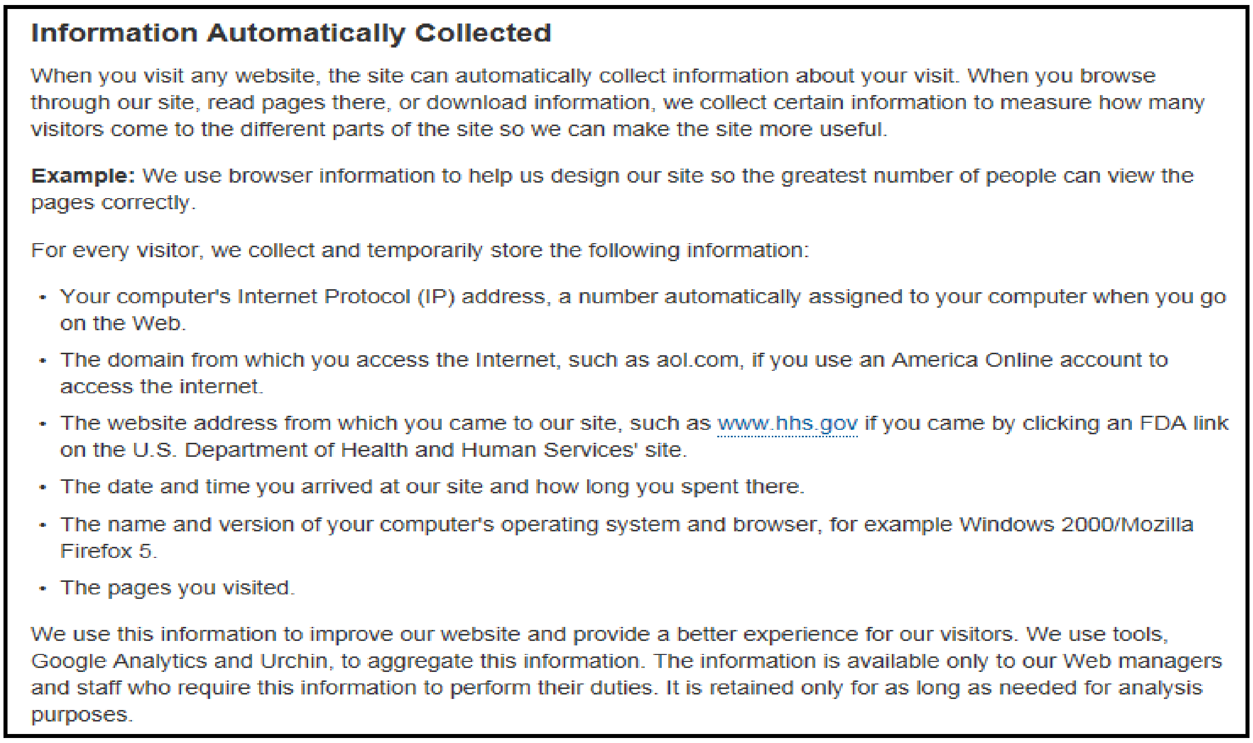
The ETL Process which includes data access from FDA Official Public Site, completely comply with Website Policies of FDA. The Policies are referred from http://www.fda.gov/AboutFDA/AboutThisWebsite/WebsitePolicies/.


U.S. Department of Health and Human Services
U.S. Food and Drug Administration
Protecting and Promoting Your Health
A to Z Index | Follow FDA | En Español
Home | Food | Drugs | Medical Devices | Radiation-Emitting Products | Vaccines, Blood & Biologics | Animal & Veterinary | Cosmetics | Tobacco Products
About FDA
Home > About FDA > About This Website > Website Policies
About This Website
Website Policies
FDA Logo Policy
Viewing Files
Website Disclaimer

Website Policies

Thanks for visiting FDA's Web site.

- Privacy Policy
- Links to Other Sites
- Third-Party Websites and Applications
- Website Content
- Information Quality
- Linking To or Copying Information On the FDA Website
- Pop-Up Advertisements
- Intrusion Detection
- Contact Information



Information Automatically Collected

When you visit any website, the site can automatically collect information about your visit. When you browse through our site, read pages there, or download information, we collect certain information to measure how many visitors come to the different parts of the site so we can make the site more useful.

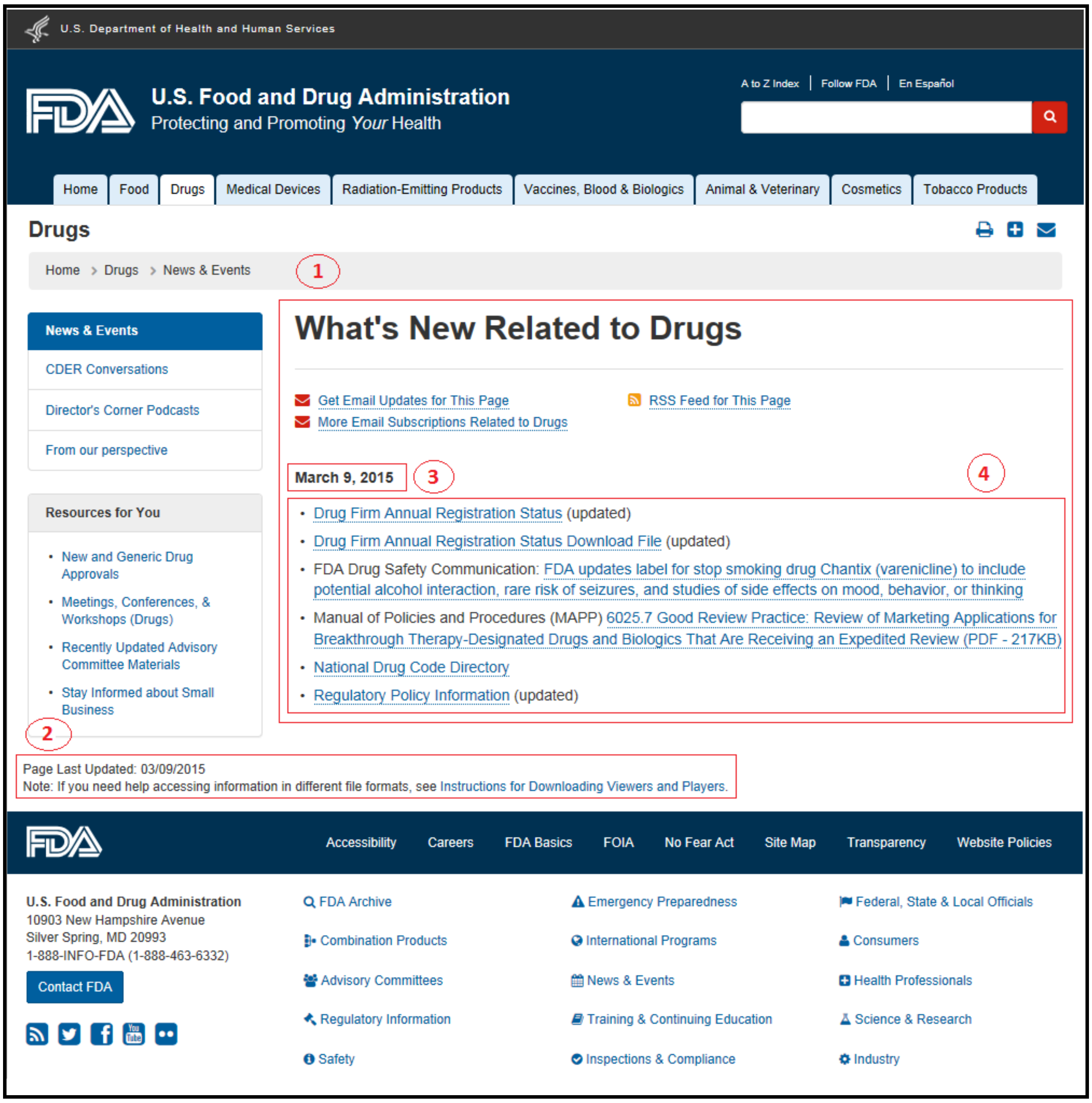
Example: We use browser information to help us design our site so the greatest number of people can view the pages correctly.

For every visitor, we collect and temporarily store the following information:

- Your computer's Internet Protocol (IP) address, a number automatically assigned to your computer when you go on the Web.
- The domain from which you access the Internet, such as aol.com, if you use an America Online account to access the internet.
- The website address from which you came to our site, such as www.hhs.gov if you came by clicking an FDA link on the U.S. Department of Health and Human Services' site.
- The date and time you arrived at our site and how long you spent there.
- The name and version of your computer's operating system and browser, for example Windows 2000/Mozilla Firefox 5.
- The pages you visited.

We use this information to improve our website and provide a better experience for our visitors. We use tools, Google Analytics and Urchin, to aggregate this information. The information is available only to our Web managers and staff who require this information to perform their duties. It is retained only for as long as needed for analysis purposes.

Figure 2-2: FDA Web Site Policies


U.S. Department of Health and Human Services

U.S. Food and Drug Administration
Protecting and Promoting *Your* Health

A to Z Index | Follow FDA | En Español

Home | Food | Drugs | Medical Devices | Radiation-Emitting Products | Vaccines, Blood & Biologics | Animal & Veterinary | Cosmetics | Tobacco Products

Drugs

Home > Drugs > News & Events 1

News & Events
CDER Conversations
Director's Corner Podcasts
From our perspective

Resources for You
- New and Generic Drug Approvals
- Meetings, Conferences, & Workshops (Drugs)
- Recently Updated Advisory Committee Materials
- Stay Informed about Small Business

What's New Related to Drugs

Get Email Updates for This Page
More Email Subscriptions Related to Drugs
RSS Feed for This Page

March 9, 2015 3

4

- Drug Firm Annual Registration Status (updated)
- Drug Firm Annual Registration Status Download File (updated)
- FDA Drug Safety Communication: FDA updates label for stop smoking drug Chantix (varenicline) to include potential alcohol interaction, rare risk of seizures, and studies of side effects on mood, behavior, or thinking
- Manual of Policies and Procedures (MAPP) 6025.7 Good Review Practice: Review of Marketing Applications for Breakthrough Therapy-Designated Drugs and Biologics That Are Receiving an Expedited Review (PDF - 217KB)
- National Drug Code Directory
- Regulatory Policy Information (updated)

2

Page Last Updated: 03/09/2015
Note: If you need help accessing information in different file formats, see Instructions for Downloading Viewers and Players.

Accessibility | Careers | FDA Basics | FOIA | No Fear Act | Site Map | Transparency | Website Policies

U.S. Food and Drug Administration
10903 New Hampshire Avenue
Silver Spring, MD 20993
1-888-INFO-FDA (1-888-463-6332)
Contact FDA

FDA Archive | Emergency Preparedness | Federal, State & Local Officials
Combination Products | International Programs | Consumers
Advisory Committees | News & Events | Health Professionals
Regulatory Information | Training & Continuing Education | Science & Research
Safety | Inspections & Compliance | Industry

Figure 2-3: FDA News & Events Page Layout

Layout of FDA News & Events Page

1 – 'What's New Related to Drugs' division is header of News & Events Page.
2 – Page Last Updated date will be available in this division.
3 – News date will be available in this division.
4 – News list with link details will be available in this division.

These are the four main elements from FDA News & Events page will be used to validate date, extract latest News and it details.

Execution flow:

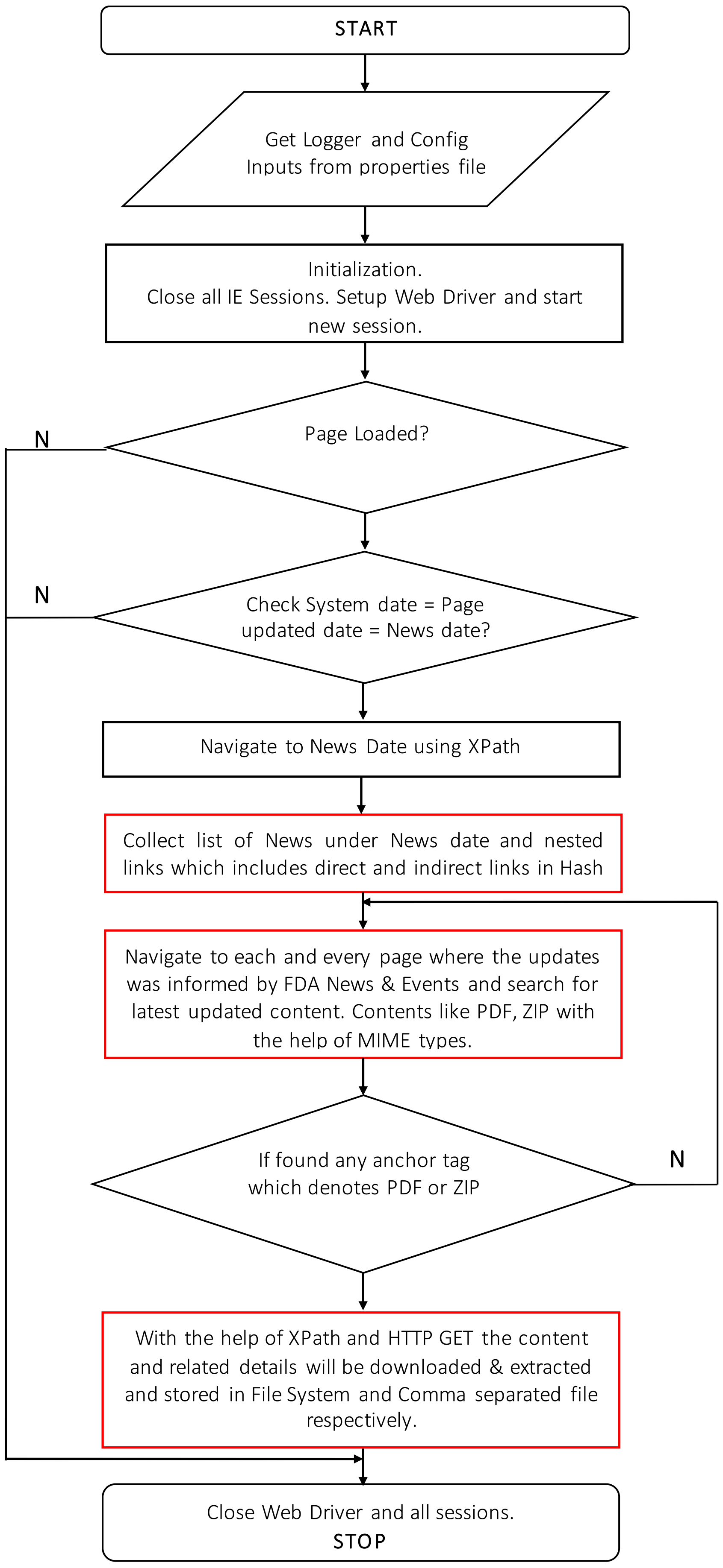
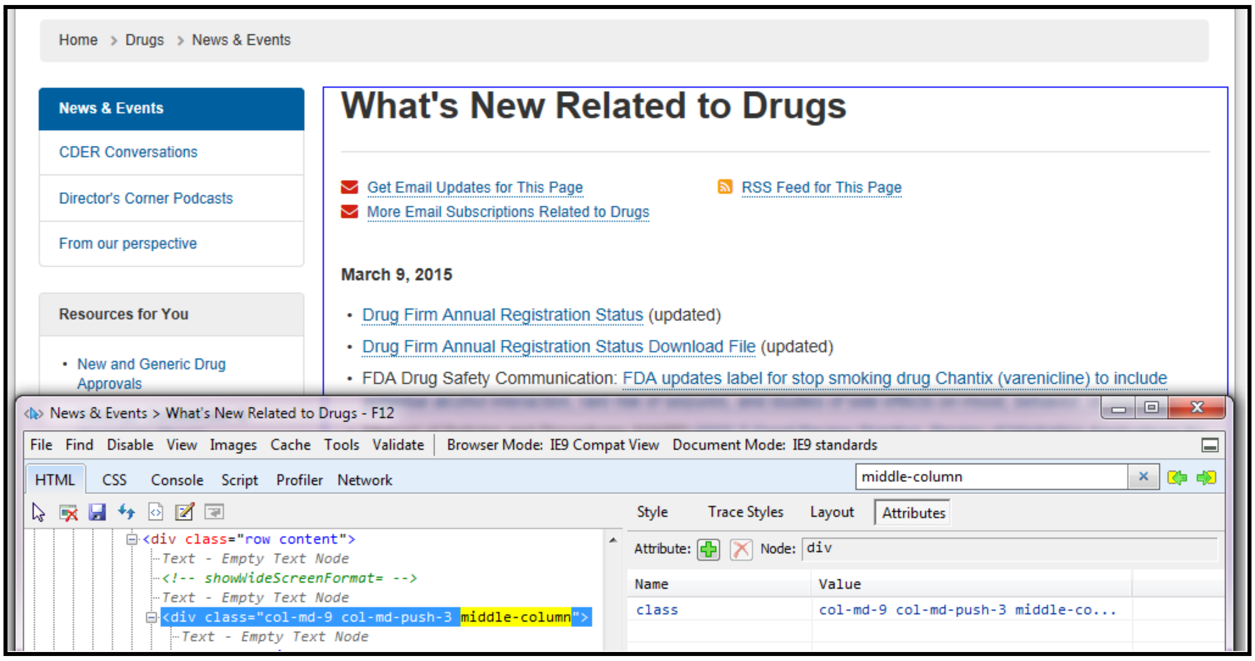

START
Get Logger and Config Inputs from properties file
Initialization. Close all IE Sessions. Setup Web Driver and start new session.
Page Loaded?
N
Check System date = Page updated date = News date?
N
Navigate to News Date using XPath
Collect list of News under News date and nested links which includes direct and indirect links in Hash
Navigate to each and every page where the updates was informed by FDA News & Events and search for latest updated content. Contents like PDF, ZIP with the help of MIME types.
If found any anchor tag which denotes PDF or ZIP
N
With the help of XPath and HTTP GET the content and related details will be downloaded & extracted and stored in File System and Comma separated file respectively.
Close Web Driver and all sessions. STOP

Figure 2-4: Extraction – Execution Flow

1. Close all IE Sessions. Setup Web Driver and start new session.
2. Check whether the FDA News & Event Page has been loaded to IE.
   By validating 'Page Last Updated' element, check whether any update is available in FDA News & Events Page. *i.e.: System date = Page updated date = News date*
3. If the Step-2 validation results as pass, then navigate for News date.
4. If the Step-3 results as pass, the list of News under that date and nested links which includes direct and indirect links will be collected.
5. Navigate to each and every page where the updates was informed by FDA News & Events and search for latest updated content. Contents like PDF, ZIP.
6. If Step-5 found any anchor tag which denotes PDF or ZIP with the help MIME type, the content and related details will be extracted and stored in File System and Comma separated file respectively. Content downloaded will the help of Http Get.
7. Close Web Driver and all sessions.



Home > Drugs > News & Events

News & Events
CDER Conversations
Director's Corner Podcasts
From our perspective
Resources for You
• New and Generic Drug Approvals

What's New Related to Drugs

Get Email Updates for This Page
RSS Feed for This Page
More Email Subscriptions Related to Drugs

March 9, 2015

• Drug Firm Annual Registration Status (updated)
• Drug Firm Annual Registration Status Download File (updated)
• FDA Drug Safety Communication: FDA updates label for stop smoking drug Chantix (varenicline) to include

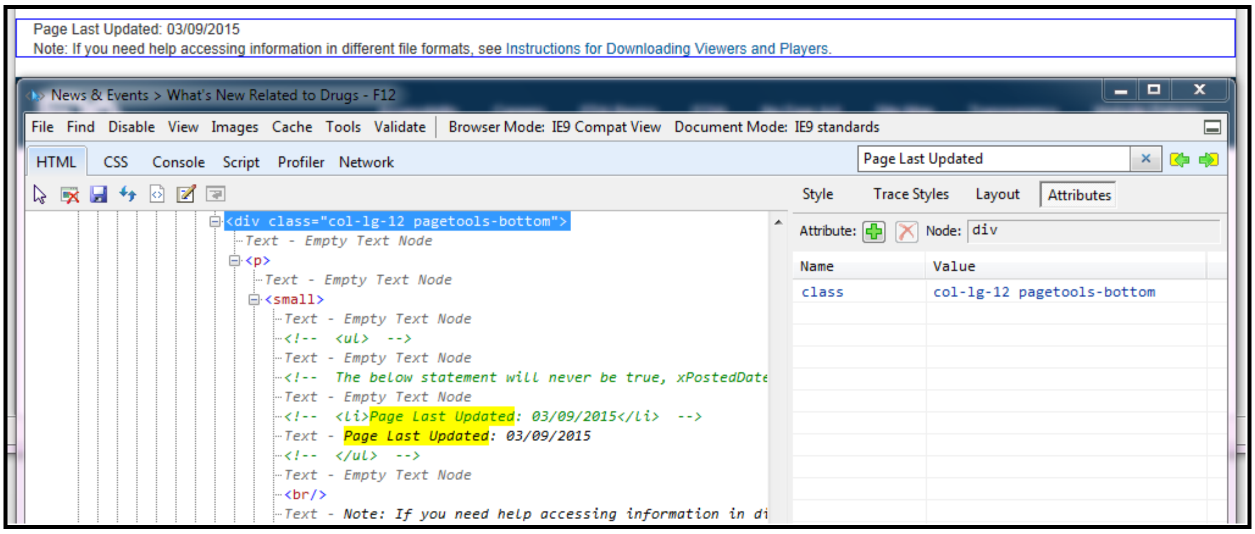
News & Events > What's New Related to Drugs - F12
File Find Disable View Images Cache Tools Validate Browser Mode: IE9 Compat View Document Mode: IE9 standards
HTML CSS Console Script Profiler Network middle-column

```
<div class="row content">
  Text - Empty Text Node
  <!-- showWideScreenFormat= -->
  Text - Empty Text Node
  <div class="col-md-9 col-md-push-3 middle-column">
    Text - Empty Text Node
```

Style Trace Styles Layout Attributes
Attribute: Node: div

Name	Value
class	col-md-9 col-md-push-3 middle-co...

Figure 2-5: XPath for Main Page Load *//div[contains(@class,'middle-column')]//p*



Page Last Updated: 03/09/2015
Note: If you need help accessing information in different file formats, see Instructions for Downloading Viewers and Players.

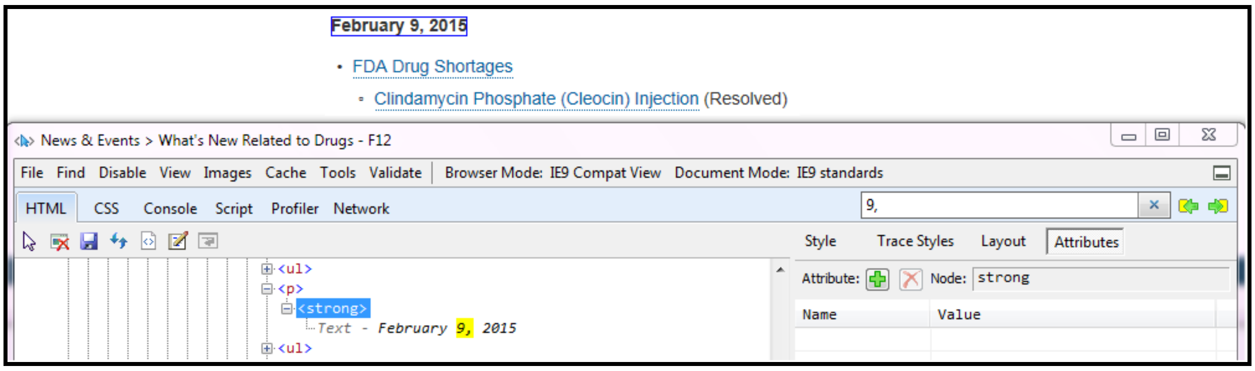
News & Events > What's New Related to Drugs - F12
File Find Disable View Images Cache Tools Validate Browser Mode: IE9 Compat View Document Mode: IE9 standards
HTML CSS Console Script Profiler Network Page Last Updated

```
<div class="col-lg-12 pagetools-bottom">
  Text - Empty Text Node
  <p>
    Text - Empty Text Node
    <small>
      Text - Empty Text Node
      <!--  <ul>  -->
      Text - Empty Text Node
      <!--  The below statement will never be true, xPostedDate
      Text - Empty Text Node
      <!--  <li>Page Last Updated: 03/09/2015</li>  -->
      Text - Page Last Updated: 03/09/2015
      <!--  </ul>  -->
      Text - Empty Text Node
      <br/>
      Text - Note: If you need help accessing information in di
```

Style Trace Styles Layout Attributes
Attribute: Node: div

Name	Value
class	col-lg-12 pagetools-bottom

Figure 2-6: XPath for Page last updated *//div[@class='col-lg-12 pagetools-bottom']/p*


February 9, 2015

- FDA Drug Shortages
  - Clindamycin Phosphate (Cleocin) Injection (Resolved)

News & Events > What's New Related to Drugs - F12

File Find Disable View Images Cache Tools Validate | Browser Mode: IE9 Compat View Document Mode: IE9 standards

HTML CSS Console Script Profiler Network

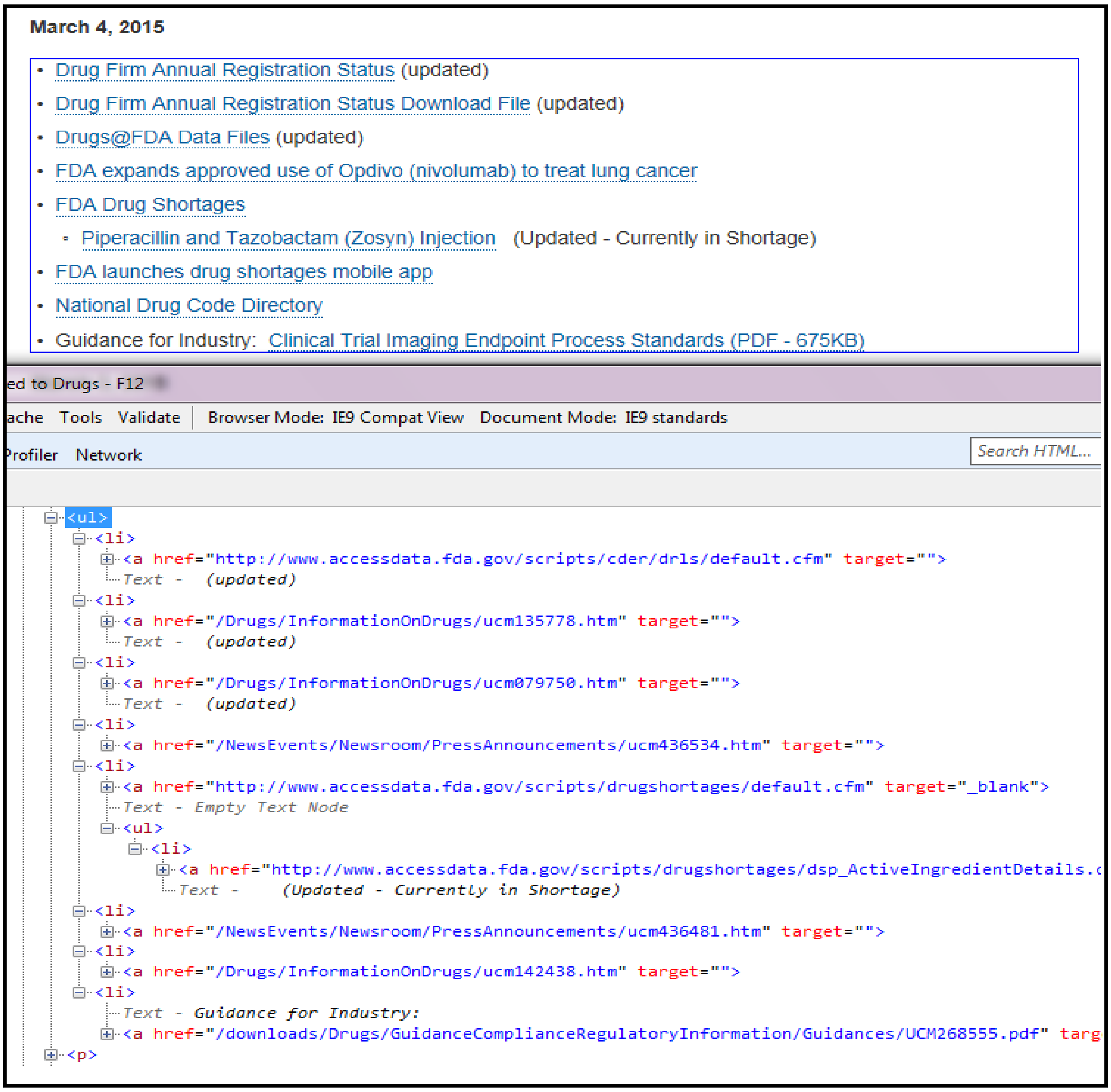
```
⊞<ul>
⊟<p>
  ⊟<strong>
     Text - February 9, 2015
⊞<ul>
```

Style Trace Styles Layout Attributes

Attribute: Node: strong

Name	Value

Figure 2-7: XPath for News date *(//div[contains(@class,'middle-column')]//p/strong)[1]*


March 4, 2015

- Drug Firm Annual Registration Status (updated)
- Drug Firm Annual Registration Status Download File (updated)
- Drugs@FDA Data Files (updated)
- FDA expands approved use of Opdivo (nivolumab) to treat lung cancer
- FDA Drug Shortages
  - Piperacillin and Tazobactam (Zosyn) Injection (Updated - Currently in Shortage)
- FDA launches drug shortages mobile app
- National Drug Code Directory
- Guidance for Industry: Clinical Trial Imaging Endpoint Process Standards (PDF - 675KB)

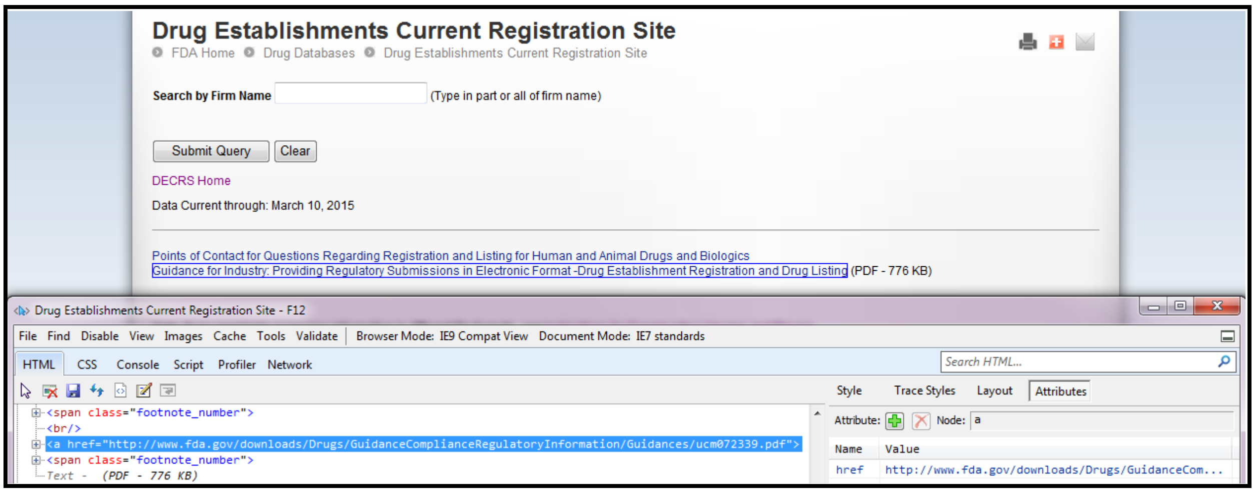
ed to Drugs - F12

ache Tools Validate | Browser Mode: IE9 Compat View Document Mode: IE9 standards

rofiler Network

```
⊟<ul>
  ⊟<li>
    ⊞<a href="http://www.accessdata.fda.gov/scripts/cder/drls/default.cfm" target="">
     Text - (updated)
  ⊟<li>
    ⊞<a href="/Drugs/InformationOnDrugs/ucm135778.htm" target="">
     Text - (updated)
  ⊟<li>
    ⊞<a href="/Drugs/InformationOnDrugs/ucm079750.htm" target="">
     Text - (updated)
  ⊟<li>
    ⊞<a href="/NewsEvents/Newsroom/PressAnnouncements/ucm436534.htm" target="">
  ⊟<li>
    ⊞<a href="http://www.accessdata.fda.gov/scripts/drugshortages/default.cfm" target="_blank">
     Text - Empty Text Node
    ⊟<ul>
      ⊟<li>
        ⊞<a href="http://www.accessdata.fda.gov/scripts/drugshortages/dsp_ActiveIngredientDetails.c
         Text -     (Updated - Currently in Shortage)
  ⊟<li>
    ⊞<a href="/NewsEvents/Newsroom/PressAnnouncements/ucm436481.htm" target="">
  ⊟<li>
    ⊞<a href="/Drugs/InformationOnDrugs/ucm142438.htm" target="">
  ⊟<li>
     Text - Guidance for Industry:
    ⊞<a href="/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/UCM268555.pdf" targ
⊞<p>
```

Figure 2-8: XPath for list of News *(//div[contains(@class,'middle-column')]//ul)[1]/li//li/a*



Drug Establishments Current Registration Site

FDA Home · Drug Databases · Drug Establishments Current Registration Site

Search by Firm Name (Type in part or all of firm name)

Submit Query | Clear

DECRS Home

Data Current through: March 10, 2015

Points of Contact for Questions Regarding Registration and Listing for Human and Animal Drugs and Biologics

Guidance for Industry: Providing Regulatory Submissions in Electronic Format -Drug Establishment Registration and Drug Listing (PDF - 776 KB)

Drug Establishments Current Registration Site - F12

File Find Disable View Images Cache Tools Validate | Browser Mode: IE9 Compat View Document Mode: IE7 standards

HTML CSS Console Script Profiler Network

```
<span class="footnote_number">
<br/>
<a href="http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/ucm072339.pdf">
<span class="footnote_number">
Text - (PDF - 776 KB)
```

Style | Trace Styles | Layout | Attributes

Attribute: Node: a

Name	Value
href	http://www.fda.gov/downloads/Drugs/GuidanceCom...

Figure 2-9: XPath to get anchor tags of PDF or ZIP from child pages *//a*

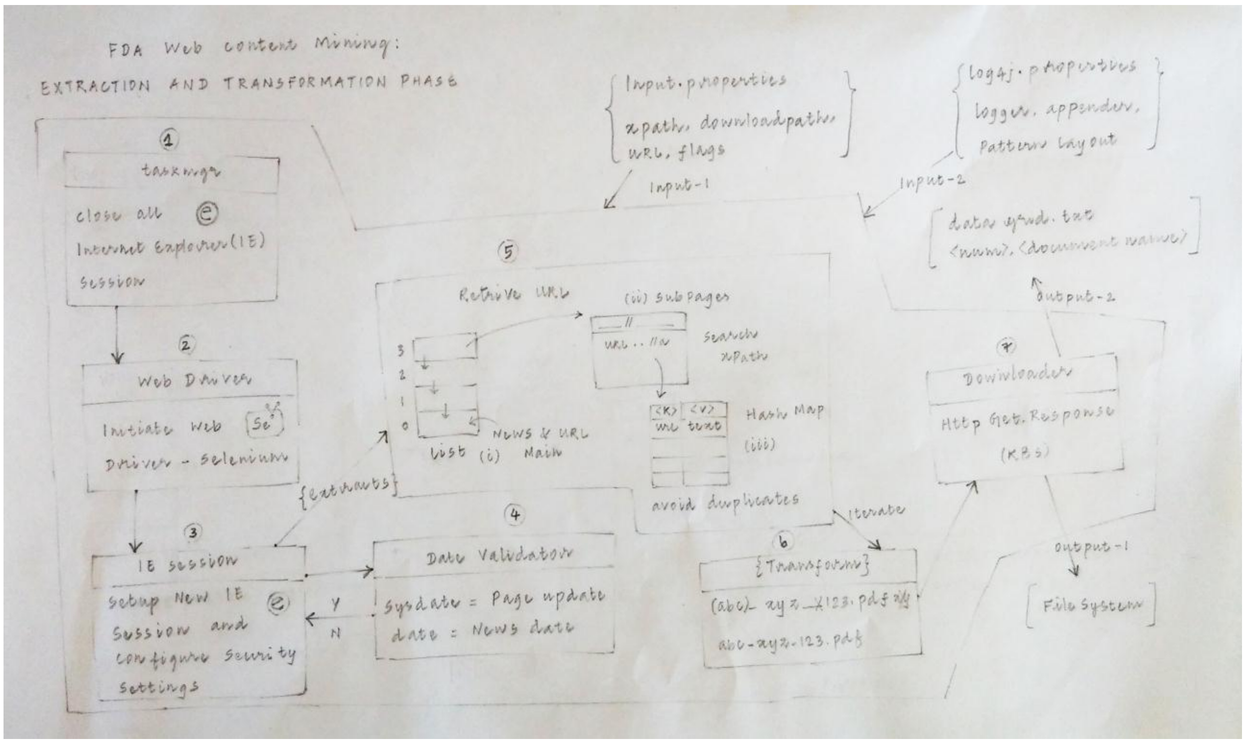


Figure 2-10: Extraction and Transformation – Technical Flow

Transformation examples of extracted data from FDA Website:

A. Document URL which is extracted from href attribute of anchor element

1. Remove unwanted texts appended at the end of href

For Example:
Extracted - http://www.fda.gov/downloads/Drugs/DevProcess/UCM378108.pdf_1
Transformed - http://www.fda.gov/downloads/Drugs/DevProcess/UCM378108.pdf

B. Full URL

2. Convert sub URLs to full URL

For Example:
Extracted - /Drugs/InformationOnDrugs/ucm135778.htm
Transformed - http://www.fda.gov/Drugs/InformationOnDrugs/ucm135778.htm

C. Document Title which is extracted from text attribute of anchor element

3. Replace all non a-z or A-Z or 0-9 texts with blank spaces
4. Replace all blank spaces with underscores
5. Remove all duplicate or continuous underscores
6. Limit no. of characters to the required length
7. After limitation, remove underscores at beginning or end of the document title

For Example:
Extracted – Drug Establishment (Annual Registration_Status) +Clinical__
Transformed - Drug_Establishment_Annual_Registration_Status

2.6 Design

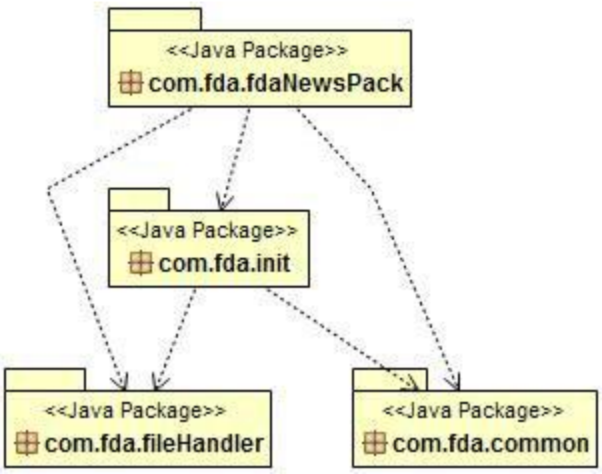
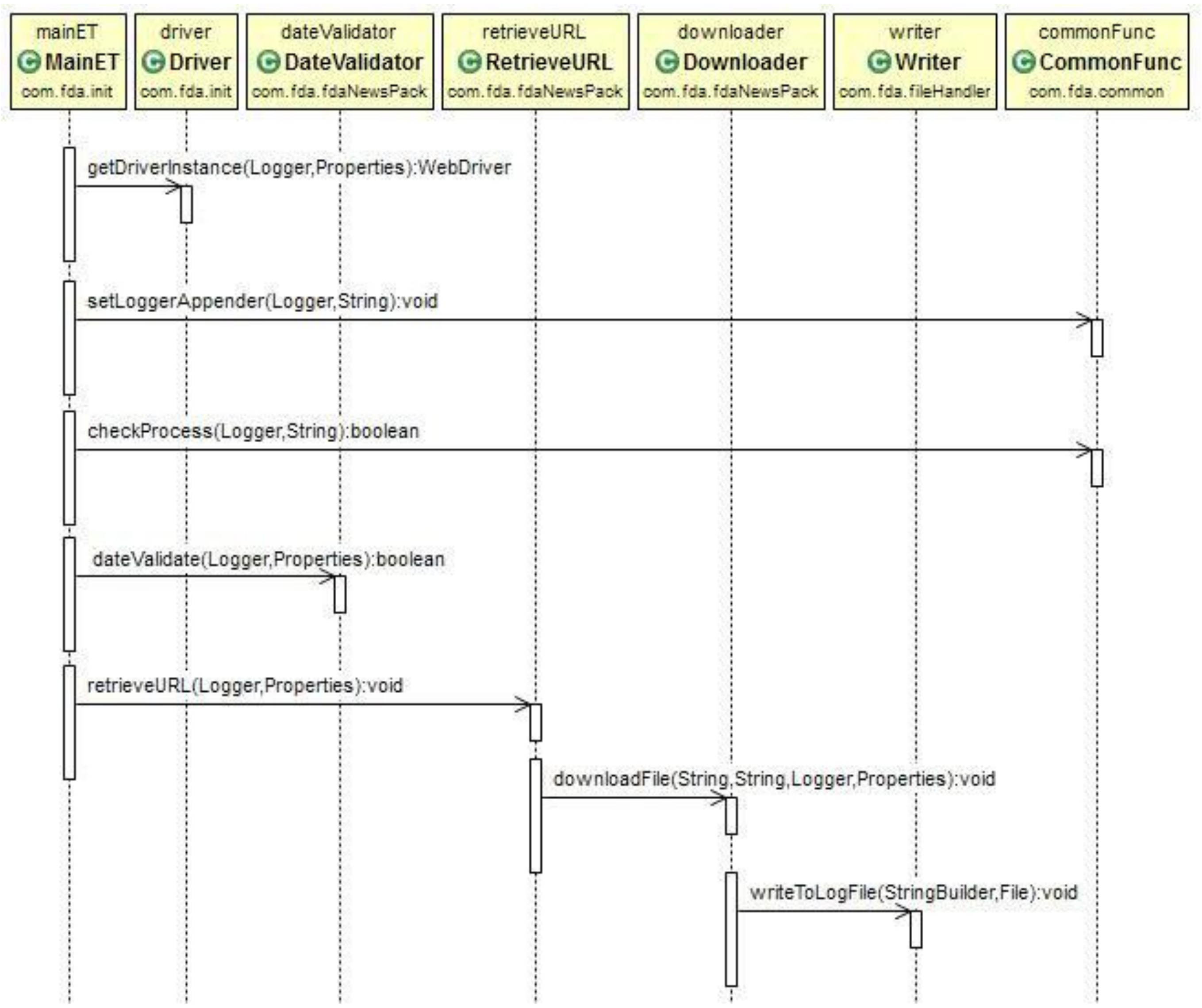
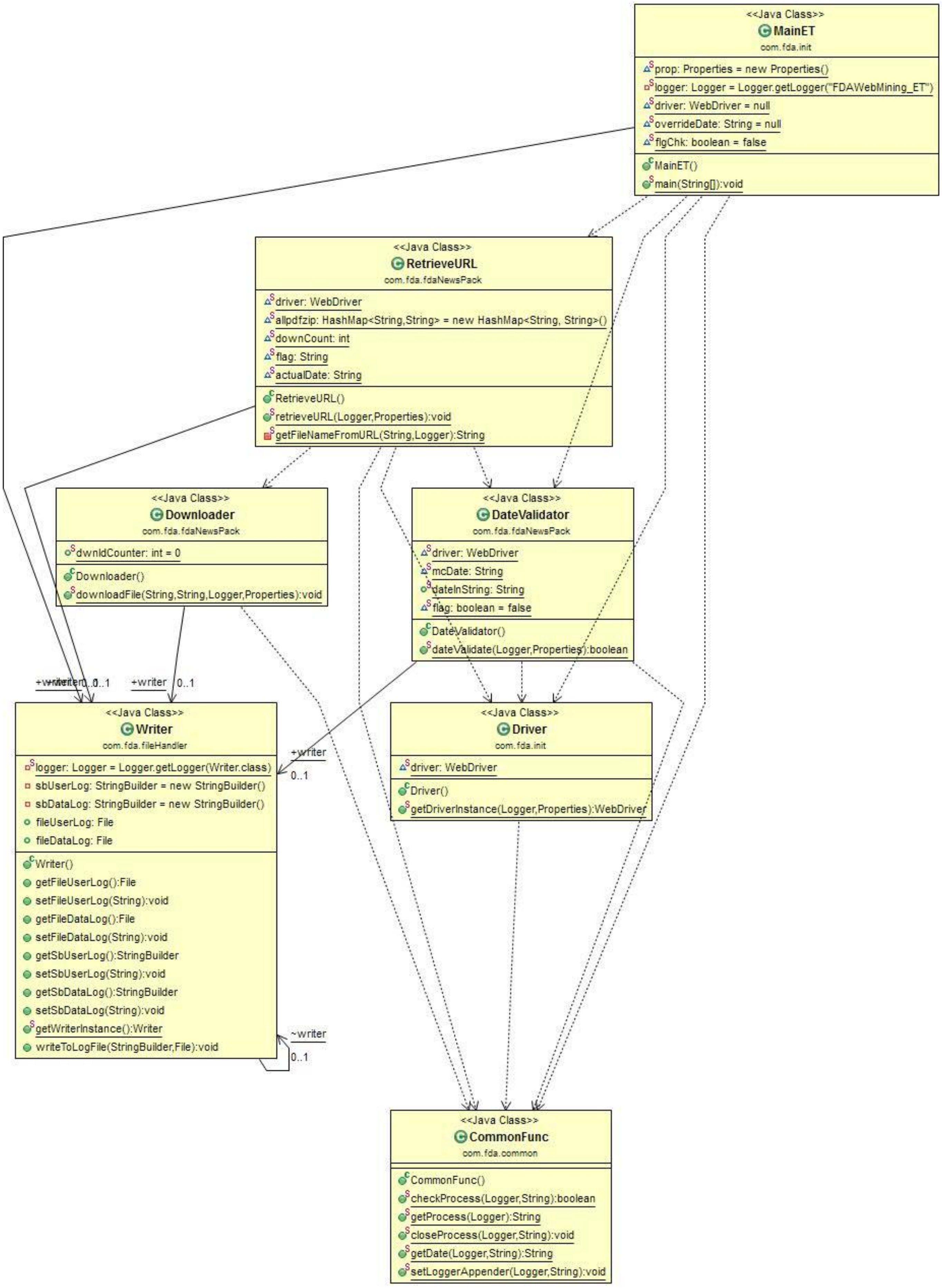


Figure 2-11: Extraction & Transformation - Package Diagram



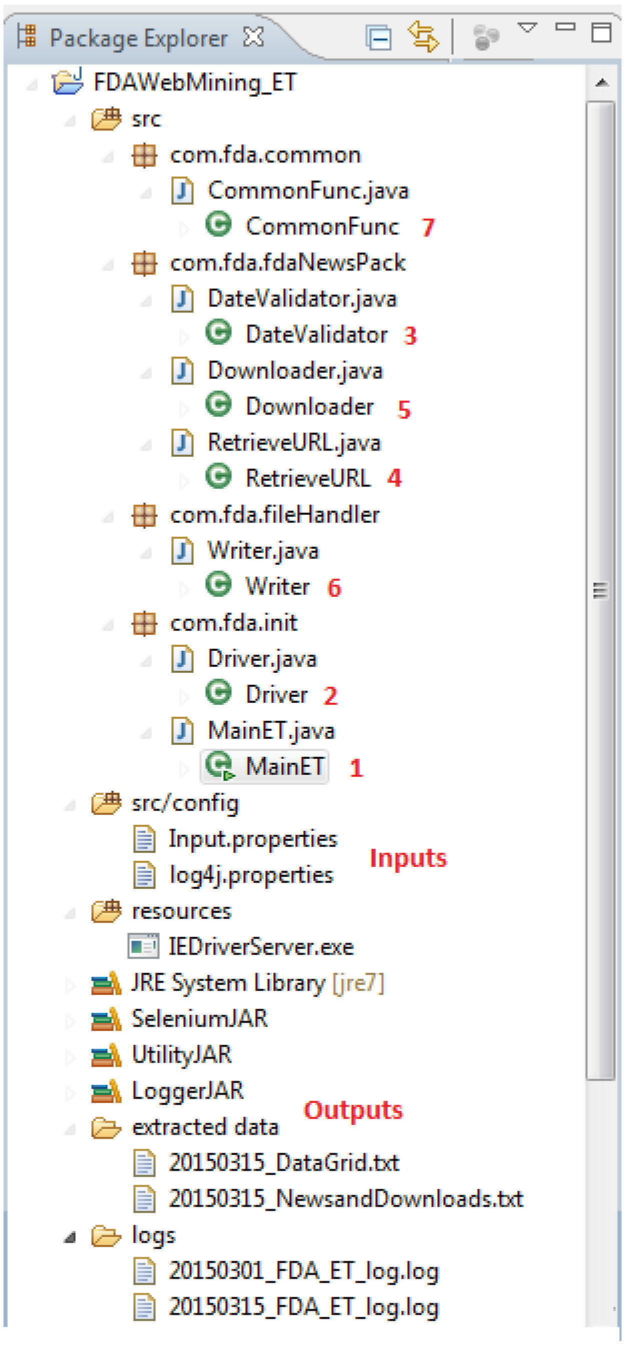
mainET
MainET
com.fda.init
driver
Driver
com.fda.init
dateValidator
DateValidator
com.fda.fdaNewsPack
retrieveURL
RetrieveURL
com.fda.fdaNewsPack
downloader
Downloader
com.fda.fdaNewsPack
writer
Writer
com.fda.fileHandler
commonFunc
CommonFunc
com.fda.common
getDriverInstance(Logger,Properties):WebDriver
setLoggerAppender(Logger,String):void
checkProcess(Logger,String):boolean
dateValidate(Logger,Properties):boolean
retrieveURL(Logger,Properties):void
downloadFile(String,String,Logger,Properties):void
writeToLogFile(StringBuilder,File):void

Figure 2-12: Extraction & Transformation - Sequence Diagram



<<Java Class>>
MainET
com.fda.init
prop: Properties = new Properties()
logger: Logger = Logger.getLogger("FDAWebMining_ET")
driver: WebDriver = null
overrideDate: String = null
flgChk: boolean = false
MainET()
main(String[]):void
<<Java Class>>
RetrieveURL
com.fda.fdaNewsPack
driver: WebDriver
allpdfzip: HashMap<String,String> = new HashMap<String, String>()
downCount: int
flag: String
actualDate: String
RetrieveURL()
retrieveURL(Logger,Properties):void
getFileNameFromURL(String,Logger):String
<<Java Class>>
Downloader
com.fda.fdaNewsPack
dwnldCounter: int = 0
Downloader()
downloadFile(String,String,Logger,Properties):void
<<Java Class>>
DateValidator
com.fda.fdaNewsPack
driver: WebDriver
mcDate: String
dateInString: String
flag: boolean = false
DateValidator()
dateValidate(Logger,Properties):boolean
+writer 0..1
+writer 0..1
+writer 0..1
<<Java Class>>
Writer
com.fda.fileHandler
logger: Logger = Logger.getLogger(Writer.class)
sbUserLog: StringBuilder = new StringBuilder()
sbDataLog: StringBuilder = new StringBuilder()
fileUserLog: File
fileDataLog: File
Writer()
getFileUserLog():File
setFileUserLog(String):void
getFileDataLog():File
setFileDataLog(String):void
getSbUserLog():StringBuilder
setSbUserLog(String):void
getSbDataLog():StringBuilder
setSbDataLog(String):void
getWriterInstance():Writer
writeToLogFile(StringBuilder,File):void
~writer
0..1
<<Java Class>>
Driver
com.fda.init
driver: WebDriver
Driver()
getDriverInstance(Logger,Properties):WebDriver
<<Java Class>>
CommonFunc
com.fda.common
CommonFunc()
checkProcess(Logger,String):boolean
getProcess(Logger):String
closeProcess(Logger,String):void
getDate(Logger,String):String
setLoggerAppender(Logger,String):void

Figure 2-13: Extraction & Transformation - Class Diagram



```
Package Explorer
FDAWebMining_ET
  src
    com.fda.common
      CommonFunc.java
        CommonFunc  7
    com.fda.fdaNewsPack
      DateValidator.java
        DateValidator  3
      Downloader.java
        Downloader  5
      RetrieveURL.java
        RetrieveURL  4
    com.fda.fileHandler
      Writer.java
        Writer  6
    com.fda.init
      Driver.java
        Driver  2
      MainET.java
        MainET  1
  src/config
    Input.properties
    log4j.properties        Inputs
  resources
    IEDriverServer.exe
  JRE System Library [jre7]
  SeleniumJAR
  UtilityJAR
  LoggerJAR
  extracted data          Outputs
    20150315_DataGrid.txt
    20150315_NewsandDownloads.txt
  logs
    20150301_FDA_ET_log.log
    20150315_FDA_ET_log.log
```

Figure 2-14: Extraction & Transformation - Framework

2.7 Code

```java
package com.fda.init;

import org.apache.log4j.Logger;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;
import java.io.InputStream;
import java.util.Date;
import java.util.Properties;
import com.fda.common.CommonFunc;
import com.fda.fdaNewsPack.DateValidator;
import com.fda.fdaNewsPack.RetrieveURL;
import com.fda.fileHandler.Writer;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * MainET: Main class of this package.
 *           All the methods will be called. Once all the methods return values, Final block will be executed.
 */
public class MainET {
     static Properties     prop          = new Properties();
     private static Logger logger        = Logger.getLogger("FDAWebMining_ET");
     static WebDriver      driver        = null;
     public static Writer  writer        = null;
     static String         overrideDate = null;
     static boolean        flgChk        = false;

     /**
      * This method is Main method of this package.
      */
     public static void main(String[] args) throws Exception {
            String propFileName = "Input.properties";
            try {
                  // Override Logger file name by prepending date
                  CommonFunc.setLoggerAppender(logger, "./logs/" + CommonFunc.getDate(logger, "yyyyMMdd")
                          + "_FDA_ET_log.log");
                  Date start = new Date();
String sDate = start.toString();
// Load Input properties file
InputStream inputStream = MainET.class.getClassLoader().getResourceAsStream(
        propFileName);
prop.load(inputStream);
// Create instance for Writer class
writer = Writer.getWriterInstance();
if (writer.getFileUserLog() == null) {
      writer.setFileUserLog(CommonFunc.getDate(logger, "yyyyMMdd") + "_"
              + prop.getProperty("fileWrite"));
}
// Create instance for Driver class
driver = Driver.getDriverInstance(logger, prop);
driver.get(prop.getProperty("url").trim());
Thread.sleep(5000);
WebDriverWait wait = new WebDriverWait(driver, 100);
// Check whether the FDA Drugs New & Events page has loaded in IE
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath(prop
        .getProperty("mainPgLoadDataXPath"))));
writer.setSbUserLog("]]]]]]]]]]]]]]]]]]]]********************[[[[[[[[[[[[[[[[[[["
        + System.getProperty("line.separator"));
writer.setSbUserLog("Title:: " + driver.getTitle()
        + System.getProperty("line.separator"));
logger.info("]]]]]]]]]]]]]]]]]]]]********************[[[[[[[[[[[[[[[[[[[");
logger.info("Title:: " + driver.getTitle());
writer.setSbUserLog(prop.getProperty("url").trim()
        + System.getProperty("line.separator"));
logger.info(prop.getProperty("url").trim());
// Validate Date
flgChk = DateValidator.dateValidate(logger, prop);
// Override Date Validator by setting return flag to true always
overrideDate = prop.getProperty("overrideDate").trim();
if (overrideDate.equals("true")) {
      flgChk = true;
      writer.setSbUserLog("-->DateValidator has been overridden<--"
              + System.getProperty("line.separator"));
      logger.warn("-->DateValidator has been overridden<--");
}
if (flgChk == true) {
      // Retrieve URLs from FDA News & Events, its sub pages
      RetrieveURL.retrieveURL(logger, prop);
```

```java
                }
                Date end = new Date();
                String eDate = end.toString();
                writer.setSbUserLog("Execution started on: " + sDate
                        + System.getProperty("line.separator"));
                logger.info("Execution started on: " + sDate);
                writer.setSbUserLog("Execution ended on: " + eDate
                        + System.getProperty("line.separator"));
                logger.info("Execution ended on: " + eDate);
            } catch (Exception e) {
                e.printStackTrace();
                logger.error("Exception:", e);
            } finally {
                // Finally, write news and extracted data to text files
                writer.setSbUserLog("--COMPLETED--");
                writer.writeToLogFile(writer.getSbUserLog(), writer.getFileUserLog());
                writer.writeToLogFile(writer.getSbDataLog(), writer.getFileDataLog());
                driver.quit();
                boolean isProcessExist = CommonFunc.checkProcess(logger, "IEDriverServer.exe");
                if (isProcessExist) {
                      CommonFunc.closeProcess(logger, "IEDriverServer.exe");
                }
                logger.info("--COMPLETED--");
            }
        }
    }
```

```java
package com.fda.init;

import java.util.Properties;
import org.apache.log4j.Logger;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.ie.InternetExplorerDriver;
import org.openqa.selenium.remote.DesiredCapabilities;
import com.fda.common.CommonFunc;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * Driver: Initiate new IE session through IE Driver.
 */
public class Driver {
     static WebDriver driver;

     /**
      * This method is to close IE sessions and initiate new IE session through IE Driver.
      *
      * @param logger
      *             contains logger object
      * @param prop
      *             contains properties object
      */
     public static WebDriver getDriverInstance(Logger logger, Properties prop) {
          try {
               if (driver == null) {
                    boolean isProcessExist = CommonFunc.checkProcess(logger, "iexplore.exe");
                    // Close all IE sessions
                    if (isProcessExist) {
                         CommonFunc.closeProcess(logger, "iexplore.exe");
                    }
                    // Set IE Driver
                    System.setProperty("webdriver.ie.driver", "resources//IEDriverServer.exe");
                    DesiredCapabilities caps = DesiredCapabilities.internetExplorer();
                    // Set IE Security Domains according to IE Driver
                    caps.setCapability(
                              InternetExplorerDriver.INTRODUCE_FLAKINESS_BY_IGNORING_SECURITY_DOMAINS,
                              true);
                    // Initiate IE Driver
                    driver = new InternetExplorerDriver(caps);
               }
          } catch (Exception e) {
               e.printStackTrace();
               logger.error("Exception: ", e);
          }
          return driver;
     }

}
```

```java
package com.fda.fdaNewsPack;

import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Properties;
import org.apache.log4j.Logger;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import com.fda.common.CommonFunc;
import com.fda.fileHandler.Writer;
import com.fda.init.Driver;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * DateValidator: Validate whether System Date = Last updated Date = News Date.
 *        This class may be overridden by Overridden flag.
 */
public class DateValidator {
     static WebDriver      driver;
     public static Writer writer;
     static String         mcDate;
     public static String dateInString;
     static boolean        flag = false;

     /**
      * This method is to validate whether System Date = Last updated Date = News Date
      *
      * @param logger
      *            contains logger object
      * @param prop
      *            contains properties object
      */
     public static boolean dateValidate(Logger logger, Properties prop) {
           try {
                 driver = Driver.getDriverInstance(logger, prop);
                 writer = Writer.getWriterInstance();
                 if (writer.getFileUserLog() == null) {
                       writer.setFileUserLog(CommonFunc.getDate(logger, "yyyyMMdd") + "_"
                                   + prop.getProperty("fileWrite"));
                 }
```

```java
// System DATE
Date todate = new Date();
SimpleDateFormat formatterto = new SimpleDateFormat("MM/dd/yyyy");
String strDate = formatterto.format(todate);
Date fortoday = formatterto.parse(strDate);
formatterto.applyPattern("MMMM dd, yyyy");
String Today = formatterto.format(fortoday);
writer.setSbUserLog("Today Date:" + Today + System.getProperty("line.separator"));
logger.info("Today Date:" + Today);
// Last Updated DATE
String lastUpdated = driver.findElement(
        By.xpath(prop.getProperty("lastUpdatedXPath").trim())).getText();
dateInString = lastUpdated.substring(19, 29);
SimpleDateFormat formatterup = new SimpleDateFormat("MM/dd/yyyy");
Date date = formatterup.parse(dateInString);
formatterup.applyPattern("MMMM dd, yyyy");
String luDate = formatterup.format(date);
writer.setSbUserLog("Last Updated Date:" + luDate
        + System.getProperty("line.separator"));
logger.info("Last Updated Date:" + luDate);
// News DATE
mcDate = driver.findElement(By.xpath(prop.getProperty("NewsDateXPath").trim()))
        .getText();
writer
        .setSbUserLog("Latest News Date:" + mcDate
                + System.getProperty("line.separator"));
logger.info("Latest News Date:" + mcDate);
// Compare System DATE and Last Updated DATE
if (Today.equalsIgnoreCase(luDate)) {
      writer.setSbUserLog("System Date and Last Updated Date MATCHED!!"
              + System.getProperty("line.separator"));
      logger.info("System Date and Last Updated Date MATCHED!!");
      // Compare Last Updated DATE and News DATE
      if (luDate.equalsIgnoreCase(mcDate)) {
            writer.setSbUserLog("Last Updated Date and Latest News Date MATCHED!!"
                    + System.getProperty("line.separator"));
            logger.info("Last Updated Date and Latest News Date MATCHED!!");
            flag = true;
      } else {
            writer.setSbUserLog("Last Updated Date and Latest News Date NOT MATCHED!!"
                    + System.getProperty("line.separator"));
```

```
                         logger.info("Last Updated Date and Latest News Date NOT MATCHED!!");
                         flag = false;
                    }
               } else {
                    writer.setSbUserLog("System Date and Last Updated Date NOT MATCHED!!"
                              + System.getProperty("line.separator"));
                    logger.info("System Date and Last Updated Date NOT MATCHED!!");
                    flag = false;
               }
          } catch (Exception e) {
               e.printStackTrace();
               logger.error("Exception: ", e);
          }
          return flag;
     }
}
```

```
package com.fda.fdaNewsPack;

import java.util.HashMap;
import java.util.List;
import java.util.Properties;
import java.util.Set;
import org.apache.log4j.Logger;
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import com.fda.common.CommonFunc;
import com.fda.fileHandler.Writer;
import com.fda.init.Driver;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * RetrieveURL: Retrieve list of URLs and nested URLs from the News and its sub pages
 */
public class RetrieveURL {
     static WebDriver                   driver;
     public static Writer               writer;
```

```
static HashMap<String, String> allpdfzip = new HashMap<String, String>();
static int                     downCount;
static String                  flag;
static String                  actualDate;

/**
 * This method is to retrieve list of URLs and nested URLs from the News and its sub pages.
 *
 * @param logger
 *            contains logger object
 * @param prop
 *            contains properties object
 */
public static void retrieveURL(Logger logger, Properties prop) {
      String replaceText;
      String replaced;
      String fileExtn;
      int lnkCounter = 0;
      String[] chkHref = null;
      String fileNameTrimmed = null;

      try {
            driver = Driver.getDriverInstance(logger, prop);
            writer = Writer.getWriterInstance();
            if (writer.getFileUserLog() == null) {
                  writer.setFileUserLog(CommonFunc.getDate(logger, "yyyyMMdd") + "_"
                              + prop.getProperty("fileWrite"));
            }
            if (writer.getFileDataLog() == null) {
                  writer.setFileDataLog(CommonFunc.getDate(logger, "yyyyMMdd") + "_"
                              + prop.getProperty("dataWrite"));
            }
            // Retrieves all links under main List
            List<WebElement> allElements = driver.findElements(By.xpath(prop.getProperty(
                        "retriveAllLinksXPath").trim()));
            // Retrieves all links under div List
            List<WebElement> divElements = driver.findElements(By.xpath(prop.getProperty(
                        "retriveDivLinksXPath").trim()));
            // Retrieves all links under sub-list
            List<WebElement> subLinkElements = driver.findElements(By.xpath(prop.getProperty(
                        "retriveSubLinksXPath").trim()));
```

```
if (!divElements.isEmpty()) {
      allElements.addAll(divElements);
}
if (!subLinkElements.isEmpty()) {
      allElements.addAll(subLinkElements);
}
writer.setSbUserLog("No. of News and Events found on " + DateValidator.mcDate + ": "
        + allElements.size() + System.getProperty("line.separator"));
logger.info("No. of News and Events found on " + DateValidator.mcDate + ": "
        + allElements.size());
int num = 1;
for (WebElement element : allElements) {
      writer
              .setSbUserLog(num + " - " + element.getText() + " ("
                      + element.getAttribute("href") + ")"
                      + System.getProperty("line.separator"));
      logger.info(num + " - " + element.getText() + " (" + element.getAttribute("href")
              + ")");
      num++;
}
writer
        .setSbUserLog("*********************************************************************"
                + System.getProperty("line.separator"));
for (WebElement element : allElements) {
      String parentLink = element.getAttribute("href");
      // PARENT
      // Checks if the link is downloadable link
      if (parentLink.contains(".pdf") | parentLink.contains(".zip")) {
            writer.setSbUserLog("| | " + element.getText() + " | |"
                    + System.getProperty("line.separator"));
            writer.setSbUserLog("Title:: " + driver.getTitle()
                    + System.getProperty("line.separator"));
            writer.setSbUserLog(element.getText() + System.getProperty("line.separator"));
            writer.setSbUserLog(element.getAttribute("href")
                    + System.getProperty("line.separator"));
            logger.info("| | " + element.getText() + " | |");
            logger.info("Title:: " + driver.getTitle());
            logger.info(element.getText());
            logger.info(element.getAttribute("href"));
            writer
                    .setSbUserLog("--------------------------------------------------------------------"
                        + System.getProperty("line.separator"));
                // Cleanup unwanted characters from href text
                replaceText = element.getText();
                replaced = replaceText.replaceAll("[^a-zA-Z0-9]", " ");
                replaced = replaced.trim().replaceAll(" ", "_");
                replaced = replaced.replaceAll("___", "_");
                replaced = replaced.replaceAll("__", "_");
                // Cleanup unwanted characters appended with href
                // Retrieve and store all links to download PDF/ZIP
                if (element.getAttribute("href").contains(".pdf")) {
                    chkHref = element.getAttribute("href").split("\\.pdf");
                    allpdfzip.put(chkHref[0] + ".pdf", replaced);
                } else if (element.getAttribute("href").contains(".zip")) {
                    chkHref = element.getAttribute("href").split("\\.zip");
                    allpdfzip.put(chkHref[0] + ".zip", replaced);
                } else {
                    allpdfzip.put(element.getAttribute("href"), replaced);
                }
                lnkCounter++;
            } else {
                writer.setSbUserLog("| | " + element.getText() + " | |"
                        + System.getProperty("line.separator"));
                logger.info("| | " + element.getText() + " | |");
                // CHILD
                // Clicks link, if it is not a direct downloadable link and navigates to child
                // window and Search for downloadable links in the opened child window
                String parentHandle = driver.getWindowHandle();
                element.sendKeys(Keys.chord(Keys.CONTROL, Keys.ENTER, Keys.TAB));
                for (String childHandle : driver.getWindowHandles()) {
                    if (!childHandle.equals(parentHandle)) {
                        driver.switchTo().window(childHandle);
                        writer.setSbUserLog("Title:: " + driver.getTitle()
                                + System.getProperty("line.separator"));
                        logger.info("Title:: " + driver.getTitle());
                        // Search anchor tags from child pages
                        List<WebElement> allLink = driver.findElements(By.xpath("//a"));
                        flag = "N";
                        for (WebElement link : allLink) {
                            if (link.getAttribute("href") != null) {
                                if (link.getAttribute("href").contains(".pdf")
                                        | link.getAttribute("href").contains(".zip")) {
                                          flag = "Y";
                                          writer.setSbUserLog(link.getText()
                                                  + System.getProperty("line.separator"));
                                          writer.setSbUserLog(link.getAttribute("href")
                                                  + System.getProperty("line.separator"));
                                          logger.info(link.getText());
                                          logger.info(link.getAttribute("href"));
                                          // Cleanup unwanted characters from href text
                                          replaceText = link.getText();
                                          replaced = replaceText.replaceAll("[^a-zA-Z0-9]", " ");
                                          replaced = replaced.trim().replaceAll(" ", "_");
                                          replaced = replaced.replaceAll("___", "_");
                                          replaced = replaced.replaceAll("__", "_");
                                          // Cleanup unwanted characters appended with href
                                          // Retrieve and store all links to download PDF/ZIP
                                          if (link.getAttribute("href").contains(".pdf")) {
                                                chkHref = link.getAttribute("href").split("\\.pdf");
                                                allpdfzip.put(chkHref[0] + ".pdf", replaced);
                                          } else if (link.getAttribute("href").contains(".zip")) {
                                                chkHref = link.getAttribute("href").split("\\.zip");
                                                allpdfzip.put(chkHref[0] + ".zip", replaced);
                                          }
                                          lnkCounter++;
                                    }
                              }
                        }
                        if (flag == "N") {
                              writer.setSbUserLog("No PDF or ZIP found!"
                                      + System.getProperty("line.separator"));
                              logger.warn("No PDF or ZIP found!");
                        }
                        writer
                                .setSbUserLog("--------------------------------------------------------"
                                        + System.getProperty("line.separator"));
                        driver.close();
                  }
            }
            driver.switchTo().window(parentHandle);
      }
}
writer.setSbUserLog("Total No. of PDF or ZIP files found: " + lnkCounter
        + System.getProperty("line.separator"));
logger.info("Total No. of PDF or ZIP files found: " + lnkCounter);

if (allpdfzip.size() > 0) {
      String path = prop.getProperty("downloadPath").trim();
      writer.setSbUserLog("Path to save files:" + path
              + System.getProperty("line.separator"));
      writer.setSbUserLog(System.getProperty("line.separator"));
      logger.info("Path to save files:" + path);
      int index = 1;
      Set<String> keyAll = allpdfzip.keySet();
      for (String Key : keyAll) {
            writer
                    .setSbUserLog("- " + index + " - "
                            + System.getProperty("line.separator"));
            logger.info(" - " + index + " - ");
            if (!allpdfzip.get(Key).toString().trim().isEmpty()) {
                  if (allpdfzip.get(Key).toString().trim().length() > Integer.parseInt(prop
                          .getProperty("maxChar"))) {
                        // Trim file length to custom length
                        fileNameTrimmed = allpdfzip.get(Key).toString().trim().substring(0,
                                Integer.parseInt(prop.getProperty("maxChar")));
                        fileNameTrimmed = fileNameTrimmed.replaceAll("_", " ");
                        fileNameTrimmed = fileNameTrimmed.trim().replaceAll(" ", "_");
                  } else {
                        fileNameTrimmed = allpdfzip.get(Key).toString().trim();
                  }
            } else {
                  String[] temp = null;
                  if (Key.contains(".pdf"))
                        temp = Key.split("\\.pdf");
                  else if (Key.contains(".zip"))
                        temp = Key.split("\\.zip");
                  temp = temp[0].split("\\/");
                  int len = temp.length;
                  fileNameTrimmed = temp[len - 1];
            }

            fileExtn = Key.substring(Key.length() - 4);
            writer.setSbDataLog(getFileNameFromURL(Key, logger) + "," + fileNameTrimmed
                    + fileExtn + System.getProperty("line.separator"));
```

```
                        // File download starts here!!
                        Downloader.downloadFile(Key, path + fileNameTrimmed + fileExtn, logger, prop);
                        index++;
                    }
                    downCount = Downloader.dwnldCounter;
                    writer.setSbUserLog(System.getProperty("line.separator")
                            + "DUPLICATE ENTRIES has been EXCLUDED by hash map"
                            + System.getProperty("line.separator"));
                    writer.setSbUserLog("Total No. of PDF or ZIP files downloaded: " + downCount
                            + System.getProperty("line.separator"));
                    logger.warn("DUPLICATE ENTRIES has been EXCLUDED by hash map");
                    logger.info("Total No. of PDF or ZIP files downloaded: " + downCount);
                    writer
                            .setSbUserLog("--------------------------------------------------------------------"
                                    + System.getProperty("line.separator"));
                } else {
                    writer.setSbUserLog("\n No PDF or ZIP files found"
                            + System.getProperty("line.separator"));
                    logger.warn("No PDF or ZIP files found");
                    writer
                            .setSbUserLog("--------------------------------------------------------------------"
                                    + System.getProperty("line.separator"));
                }
            } catch (Exception e) {
                e.printStackTrace();
                logger.error("Exception:", e);
            }
        }

        /**
         * This method is to get file name from URL.
         *
         * @param url
         *            contains URL
         * @param logger
         *            contains logger object
         */
        private static String getFileNameFromURL(String url, Logger logger) {
            String fileNameFromURL = null;
            String[] temp = null;
            try {
                  if (url.contains(".pdf")) {
                        temp = url.split("\\.pdf");
                  } else if (url.contains(".zip")) {
                        temp = url.split("\\.zip");
                  }
                  temp = temp[0].split("\\/");
                  int len = temp.length;
                  fileNameFromURL = temp[len - 1];
            } catch (Exception e) {
                  e.printStackTrace();
                  logger.error("Exception:", e);
            }
            return fileNameFromURL;
      }
      }
```

```
package com.fda.fdaNewsPack;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.util.Properties;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.impl.client.HttpClientBuilder;
import org.apache.log4j.Logger;
import com.fda.common.CommonFunc;
import com.fda.fileHandler.Writer;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * Downloader: Using Http Get & Response, the FDA page status will be checked and download file in bytes.
 */
public class Downloader {
     public static Writer writer;
```

```java
public static int    dwnldCounter = 0;

/**
 * This method is to download file for the specified URL.
 *
 * @param downloadUrl
 *            contains URL which specifies from where the file to download
 * @param outputFilePath
 *            contains a path which specifies where the downloaded file to reside
 * @param logger
 *            contains logger object
 * @param prop
 *            contains properties object
 */
public static void downloadFile(String downloadUrl, String outputFilePath, Logger logger,
        Properties prop) throws IOException {
      try {
            writer = Writer.getWriterInstance();
            if (writer.getFileUserLog() == null) {
                  writer.setFileUserLog(CommonFunc.getDate(logger, "yyyyMMdd") + "_"
                              + prop.getProperty("fileWrite"));
            }
            // Http Get & Response
            HttpClient httpClient = HttpClientBuilder.create().build();
            HttpGet httpGet = new HttpGet(downloadUrl);
            writer.setSbUserLog("Downloading file from: " + downloadUrl
                        + System.getProperty("line.separator"));
            logger.info("Downloading file from: " + downloadUrl);
            HttpResponse response = httpClient.execute(httpGet);
            if (response.getStatusLine().toString().contains("OK")) {
                  writer.setSbUserLog(response.getStatusLine().toString()
                              + System.getProperty("line.separator"));
                  logger.info(response.getStatusLine().toString());
                  HttpEntity entity = response.getEntity();
                  if (entity != null) {
                        File chckDir = new File(prop.getProperty("downloadPath"));
                        // If directory does not exists, creates new directory
                        if (!chckDir.exists()) {
                              chckDir.mkdir();
                        }
                        File outputFile = new File(outputFilePath);
```

```java
                            InputStream inputStream = entity.getContent();
                            FileOutputStream fileOutputStream = new FileOutputStream(outputFile);
                            int read = 0;
                            byte[] bytes = new byte[81920000];
                            // Download file in bytes
                            while ((read = inputStream.read(bytes)) != -1) {
                                  fileOutputStream.write(bytes, 0, read);
                            }
                            fileOutputStream.close();
                            writer.setSbUserLog("Downloded " + outputFile.length() + " bytes. "
                                    + entity.getContentType() + System.getProperty("line.separator"));
                            logger.info("Downloded " + outputFile.length() + " bytes. "
                                    + entity.getContentType());
                            Downloader.dwnldCounter++;
                      } else {
                            writer.setSbUserLog("Download failed! -->" + downloadUrl
                                    + System.getProperty("line.separator"));
                            logger.warn("Download failed! -->" + downloadUrl);
                      }
                } else {
                      writer.setSbUserLog(response.getStatusLine().toString()
                              + System.getProperty("line.separator"));
                      logger.info(response.getStatusLine().toString());
                }
          } catch (Exception e) {
                e.printStackTrace();
                logger.error("Exception: ", e);
          }
    }
}
```

```java
package com.fda.fileHandler;

import java.io.*;
import org.apache.log4j.Logger;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * Writer: This class write News and Extracted data to a specified text file.
 *         User Log has News. Data Log has extracted File name and Title.
 */
public class Writer {
     static Writer         writer;
     private static Logger logger    = Logger.getLogger(Writer.class);
     private StringBuilder sbUserLog = new StringBuilder();
     private StringBuilder sbDataLog = new StringBuilder();
     public File           fileUserLog;
     public File           fileDataLog;

     /**
      * This method is FILE GETTER for USER log.
      */
     public File getFileUserLog() {
           return fileUserLog;
     }

     /**
      * This method is FILE SETTER for USER log.
      */
     public void setFileUserLog(String fileName) {
           File file = new File("extracted data\\" + fileName);
           file.getParentFile().mkdir();
           try {
                 file.createNewFile();
           } catch (IOException e) {
                 e.printStackTrace();
           }
           this.fileUserLog = file;
     }

     /**
      * This method is FILE GETTER for DATA log.
      */
     public File getFileDataLog() {
           return fileDataLog;
     }

     /**
      * This method is FILE SETTER for DATA log.
      *
```

```
	 * @param fileName
	 *            contains file name where the data needs to be go for Loading
	 */
	public void setFileDataLog(String fileName) {
		File file = new File("extracted data\\" + fileName);
		file.getParentFile().mkdir();
		try {
			file.createNewFile();
		} catch (IOException e) {
			e.printStackTrace();
		}
		this.fileDataLog = file;
	}

	/**
	 * This method is STRING BUILDER GETTER for USER log.
	 */
	public StringBuilder getSbUserLog() {
		return sbUserLog;
	}

	/**
	 * This method is STRING BUILDER SETTER for USER log.
	 */
	public void setSbUserLog(String sbUserLog) {
		this.sbUserLog.append(sbUserLog);
	}

	/**
	 * This method is STRING BUILDER GETTER for DATA log.
	 */
	public StringBuilder getSbDataLog() {
		return sbDataLog;
	}

	/**
	 * This method is STRING BUILDER SETTER for DATA log.
	 *
	 * @param sbDataLog
	 *            contains String Builder object
	 */
	public void setSbDataLog(String sbDataLog) {
		this.sbDataLog.append(sbDataLog);
	}

	/**
	 * This method is to get Writer class instance.
	 */
	public static Writer getWriterInstance() {
		if (writer == null) {
			writer = new Writer();
		}
		return writer;
	}

	/**
	 * This method is to write message to specified file.
	 *
	 * @param sbLog
	 *            contains String Builder object
	 * @param file
	 *            contains file name where user or data needs to be logged
	 */
	public void writeToLogFile(StringBuilder sbLog, File file) {
		try {
			BufferedWriter bwr = new BufferedWriter(new FileWriter(file));
			bwr.write(sbLog.toString());
			bwr.flush();
			bwr.close();
		} catch (IOException e) {
			e.printStackTrace();
			logger.error("IOException:", e);
		}
	}
}
```

```
package com.fda.common;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
```

```java
import java.util.Date;
import java.text.ParseException;
import java.text.SimpleDateFormat;

import org.apache.log4j.Logger;
import org.apache.log4j.PatternLayout;
import org.apache.log4j.RollingFileAppender;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * CommonFunc: Common Functions like check process, format date, override logger
 */
public class CommonFunc {

     /**
      * This method is to check the status of a process.
      *
      * @param logger
      *            contains logger object
      * @param ProcessName
      *            contains process name
      */
     public static boolean checkProcess(Logger logger, String ProcessName) {
           if (getProcess(logger).contains(ProcessName)) {
                 return true;
           }
           return false;
     }

     /**
      * This method is to get the list of processes and its status from task manager.
      *
      * @param logger
      *            contains logger object
      */
     public static String getProcess(Logger logger) {
           String line;
           String processInfo = "";

           Process p;
```

```
        try {
            p = Runtime.getRuntime()
                    .exec(System.getenv("windir") + "\\system32\\" + "tasklist.exe");
            BufferedReader input = new BufferedReader(new InputStreamReader(p.getInputStream()));
            while ((line = input.readLine()) != null) {
                processInfo += line;
            }
            input.close();
        } catch (IOException e) {
            e.printStackTrace();
            logger.error("Exception: ", e);
        }
        return processInfo;
}

/**
 * This method is to kill a process.
 *
 * @param logger
 *            contains logger object
 * @param processName
 *            contains process name
 */
public static void closeProcess(Logger logger, String processName) {
        try {
            Runtime.getRuntime().exec("taskkill /F /IM " + processName);
        } catch (IOException e) {
            e.printStackTrace();
            logger.error("Exception: ", e);
        }
}

/**
 * This method is to format current date string.
 *
 * @param logger
 *            contains logger object
 * @param format
 *            contains a date format
 */
public static String getDate(Logger logger, String format) throws ParseException {
```

```
            String nowDate = null;
            try {
                  SimpleDateFormat sdfDate = new SimpleDateFormat(format);// dd/MM/yyyy
                  Date now = new Date();
                  nowDate = sdfDate.format(now);
            } catch (Exception e) {
                  e.printStackTrace();
                  logger.error("Exception: ", e);
            }
            return nowDate;
      }

      /**
       * This method is to override FILE appender of logger.
       *
       * @param logger
       *             contains logger object
       * @param fName
       *             contains logger file name
       */
      public static void setLoggerAppender(Logger logger, String fName) {
            PatternLayout layout = new PatternLayout("%d{yyyy.MM.dd HH:mm:ss} - %5p [%F:%L]: %m%n");
            RollingFileAppender appender;
            try {
                  appender = new RollingFileAppender(layout, fName, false);
                  logger.addAppender(appender);
            } catch (IOException e) {
                  e.printStackTrace();
                  logger.error("Exception: ", e);
            }

      }
}
```

```
#log4j.properties#

#Log levels#
#TRACE<DEBUG<INFO<WARN<ERROR<FATAL
log4j.rootLogger=INFO,CONSOLE,R
#
#CONSOLE Appender#
log4j.appender.CONSOLE=org.apache.log4j.ConsoleAppender
#
#Pattern to output the caller's file name and line number#
log4j.appender.CONSOLE.layout=org.apache.log4j.PatternLayout
log4j.appender.CONSOLE.layout.ConversionPattern=%5p [%t] (%F:%L) - %m%n
#
#ROLLING FILE Appender#
log4j.appender.R=org.apache.log4j.rolling.RollingFileAppender
#log4j.appender.R=org.apache.log4j.RollingFileAppender
#
#Path and file name to store the log file#
log4j.appender.R.RollingPolicy=org.apache.log4j.rolling.TimeBasedRollingPolicy
log4j.appender.R.RollingPolicy.FileNamePattern=/logs/%d{yyyyMMdd}_log.log
log4j.appender.R.Append=true
#log4j.appender.R.File=./logs/log4j.log
#log4j.appender.R.MaxFileSize=200KB
#
#Number of backup files#
#log4j.appender.R.MaxBackupIndex=2
#
#Layout for Rolling File Appender#
log4j.appender.R.layout=org.apache.log4j.PatternLayout
log4j.appender.R.layout.ConversionPattern=%d{yyyy.MM.dd HH:mm:ss} - %5p [%F:%L]: %m%n
#log4j.appender.R.layout.ConversionPattern=%d - %c - %p - %m%n
```

```
#Input.properties#

#FDA New & Events Site#
url=http://www.fda.gov/Drugs/NewsEvents/ucm130958.htm
#
#Check condition for complete main page load#
mainPgLoadDataXPath=//div[contains(@class,'middle-column')]//p
#
#Last updated date from bottom left corner#
lastUpdatedXPath=//div[@class='col-lg-12 pagetools-bottom']/p
#
#News date from top middle#
NewsDateXPath=(//div[contains(@class,'middle-column')]//p/strong)[1]
#
#List of News#
retriveAllLinksXPath=(//div[contains(@class,'middle-column')]//ul)[1]/li/a
retriveDivLinksXPath=(//div[contains(@class,'middle-column')]//ul)[1]/li/div/a
retriveSubLinksXPath=(//div[contains(@class,'middle-column')]//ul)[1]/li//li/a
#
#Path for the files to be downloaded#
downloadPath=D:\\Sri Kumaran Cabinet\\BITS WILP\\Fourth Semester\\My Project\\CODE\\workspace_web_mining\\File Download\\
#
#File to track complete information of execution#
fileWrite=NewsandDownloads.txt
dataWrite=DataGrid.txt
#
#File name maximum length#
maxChar=60
#
#Override the condition 'System Date==Last Updated Date==News Date'#
overrideDate=true
```

2.8 Test Cases

Unit Testing:

Note: The below test cases should be executed for the given Input properties file. The values should not be changed. *For Example: All the Xpath has been set to 1, which reads data from index 1 of News and Events Page class = middle, i.e.: Latest. No. of characters in Document Name has been restricted to 60.*

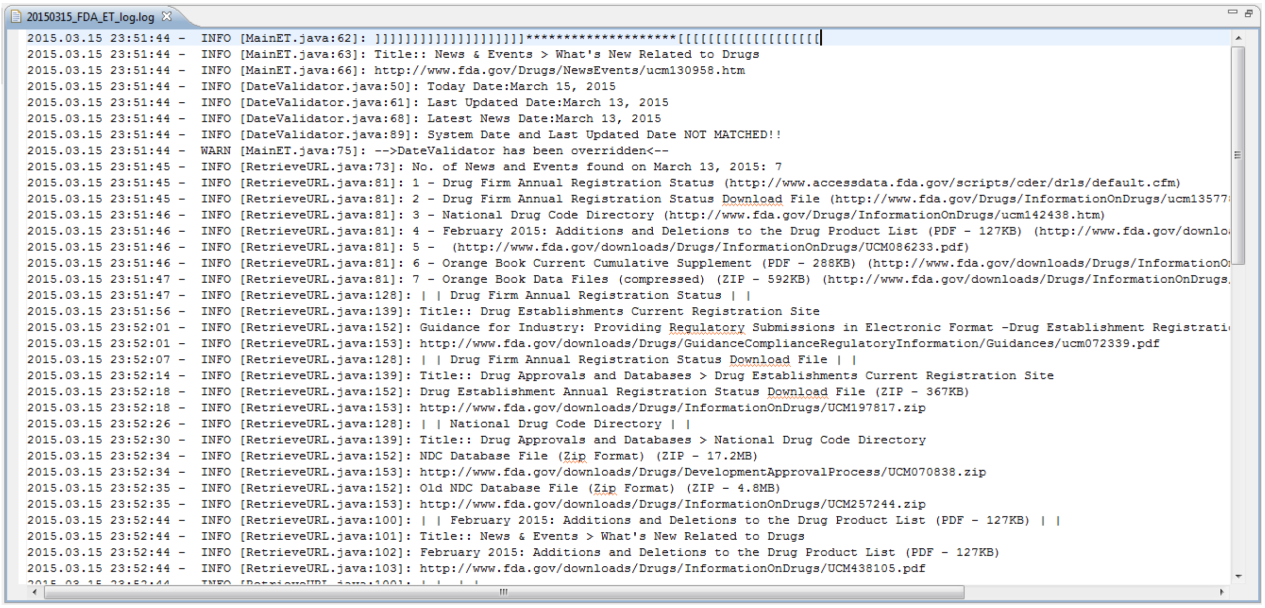
Test Case ID	Test Case	Expected	Actual	Result
1.1	Driver	At the beginning of run, close all IE Sessions if any opened.	Same	Pass
1.2		Web Driver to open port and set session between client and IE.	Same	Pass
1.3		Switch off all Pop up and reset all security settings.	Same	Pass
1.4		Launch FDA News & Events page http://www.fda.gov/Drugs/NewsEvents/ucm130958.htm and timeout should not occur till the page loads in IE.	Same	Pass
2.1	Date Validator	Execution should continue only if System Date = Last updated Date = News Date OR overrideDate flag is set to true.	Same	Pass
3.1	Retrieve URL	List of retrieved News and Events count should match with manual count from the FDA Page. This includes URLs from List, Div and sub List.	Same	Pass
3.2		If News & Events count more than one and has indirect URLs, the tool should navigate to child pages (one – by - one) and retrieve only PDF and ZIP files.	Same	Pass
3.3		Retrived URL should not include duplicates in download list.	Same	Pass
4.1	Downloader	Download file to specified folder. Download should be byte by byte.	Same	Pass
5.1	Writer	Writes News & Events and extracted data information to specified text files.	Same	Pass
5.2		Writes should executes only at successful completion of all operations.	Same	Pass
6.1	Extracted Data	News & Downloads and Grid data files should be created under "extracted data" folder.	Same	Pass
6.2		News & Downloads and Grid data files should be created and file name appended with today's date.	Same	Pass
6.3		News & Downloads file should include all today's News with URL details.	Same	Pass
6.4		Document Number and Document Name in Data Grid file should be seperated by comma.	Same	Pass
6.5		No. of characters in document name should be limited to no. of characters restricted via input properties.	Same	Pass

6.6		Document Name should not contain unwanted texts.	Same	Pass
6.7		No. of files count in News & Downloads and Data Grid files should be same.	Same	Pass
6.8		No. of file count should be match with files in "download file" folder.	Same	Pass
7.1	Log File	Log file should be created under "extracted folder".	Same	Pass
7.2		Log file should be created and file name appended with today's date.	Same	Pass
7.3		Log file information structure should be adhere to pattern defined in log4j properties.	Same	Pass
7.4		Log file should include Date, Time, Log classification, Class name and then information.	Same	Pass

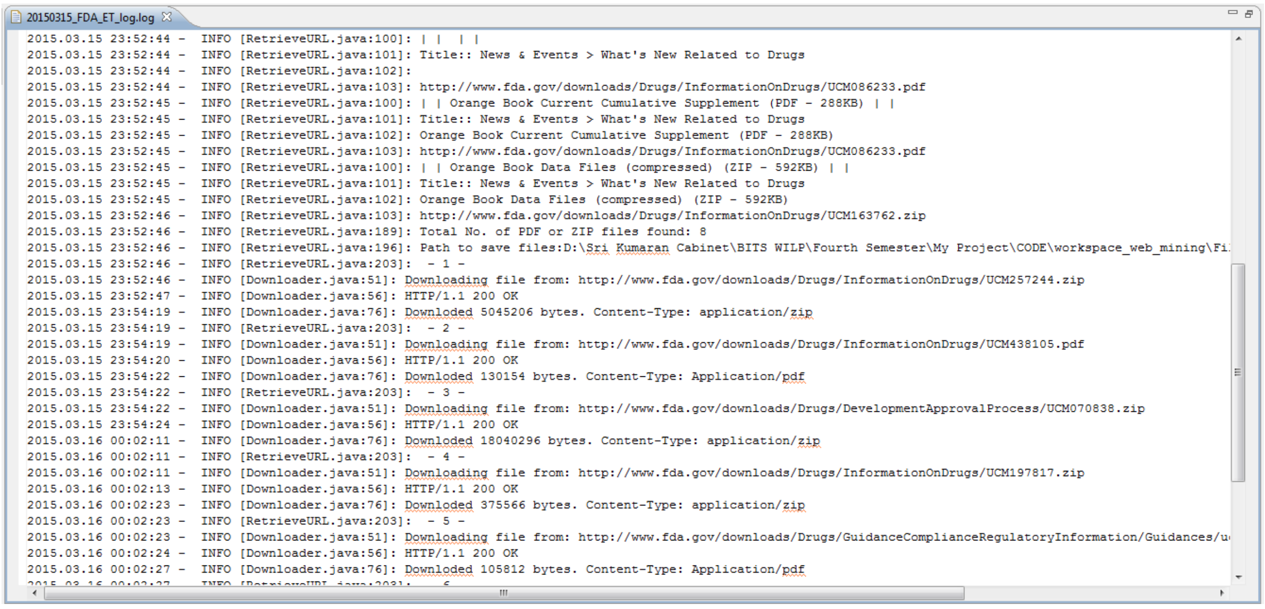
Cases:	1.1 – 7.4	Result:	PASS
Executed By:	R. Sathish Kumar	Date:	10-MAR-2015

Output:

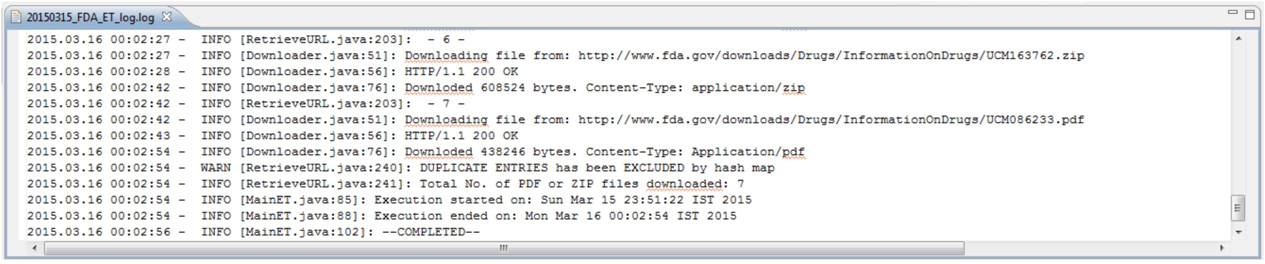
20150315_FDA_ET_log.log



```
2015.03.15 23:51:44 -  INFO [MainET.java:62]: ]]]]]]]]]]]]]]]]]]]*******************[[[[[[[[[[[[[[[[[[
2015.03.15 23:51:44 -  INFO [MainET.java:63]: Title:: News & Events > What's New Related to Drugs
2015.03.15 23:51:44 -  INFO [MainET.java:66]: http://www.fda.gov/Drugs/NewsEvents/ucm130958.htm
2015.03.15 23:51:44 -  INFO [DateValidator.java:50]: Today Date:March 15, 2015
2015.03.15 23:51:44 -  INFO [DateValidator.java:61]: Last Updated Date:March 13, 2015
2015.03.15 23:51:44 -  INFO [DateValidator.java:68]: Latest News Date:March 13, 2015
2015.03.15 23:51:44 -  INFO [DateValidator.java:89]: System Date and Last Updated Date NOT MATCHED!!
2015.03.15 23:51:44 -  WARN [MainET.java:75]: -->DateValidator has been overridden<--
2015.03.15 23:51:45 -  INFO [RetrieveURL.java:73]: No. of News and Events found on March 13, 2015: 7
2015.03.15 23:51:45 -  INFO [RetrieveURL.java:81]: 1 - Drug Firm Annual Registration Status (http://www.accessdata.fda.gov/scripts/cder/drls/default.cfm)
2015.03.15 23:51:45 -  INFO [RetrieveURL.java:81]: 2 - Drug Firm Annual Registration Status Download File (http://www.fda.gov/Drugs/InformationOnDrugs/ucm13577
2015.03.15 23:51:46 -  INFO [RetrieveURL.java:81]: 3 - National Drug Code Directory (http://www.fda.gov/Drugs/InformationOnDrugs/ucm142438.htm)
2015.03.15 23:51:46 -  INFO [RetrieveURL.java:81]: 4 - February 2015: Additions and Deletions to the Drug Product List (PDF - 127KB) (http://www.fda.gov/downlo
2015.03.15 23:51:46 -  INFO [RetrieveURL.java:81]: 5 -  (http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM086233.pdf)
2015.03.15 23:51:46 -  INFO [RetrieveURL.java:81]: 6 - Orange Book Current Cumulative Supplement (PDF - 288KB) (http://www.fda.gov/downloads/Drugs/InformationO
2015.03.15 23:51:47 -  INFO [RetrieveURL.java:81]: 7 - Orange Book Data Files (compressed) (ZIP - 592KB) (http://www.fda.gov/downloads/Drugs/InformationOnDrugs
2015.03.15 23:51:47 -  INFO [RetrieveURL.java:128]: | | Drug Firm Annual Registration Status | |
2015.03.15 23:51:56 -  INFO [RetrieveURL.java:139]: Title:: Drug Establishments Current Registration Site
2015.03.15 23:52:01 -  INFO [RetrieveURL.java:152]: Guidance for Industry: Providing Regulatory Submissions in Electronic Format -Drug Establishment Registrati
2015.03.15 23:52:01 -  INFO [RetrieveURL.java:153]: http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/ucm072339.pdf
2015.03.15 23:52:07 -  INFO [RetrieveURL.java:128]: | | Drug Firm Annual Registration Status Download File | |
2015.03.15 23:52:14 -  INFO [RetrieveURL.java:139]: Title:: Drug Approvals and Databases > Drug Establishments Current Registration Site
2015.03.15 23:52:18 -  INFO [RetrieveURL.java:152]: Drug Establishment Annual Registration Status Download File (ZIP - 367KB)
2015.03.15 23:52:18 -  INFO [RetrieveURL.java:153]: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM197817.zip
2015.03.15 23:52:26 -  INFO [RetrieveURL.java:128]: | | National Drug Code Directory | |
2015.03.15 23:52:30 -  INFO [RetrieveURL.java:139]: Title:: Drug Approvals and Databases > National Drug Code Directory
2015.03.15 23:52:34 -  INFO [RetrieveURL.java:152]: NDC Database File (Zip Format) (ZIP - 17.2MB)
2015.03.15 23:52:34 -  INFO [RetrieveURL.java:153]: http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/UCM070838.zip
2015.03.15 23:52:35 -  INFO [RetrieveURL.java:152]: Old NDC Database File (Zip Format) (ZIP - 4.8MB)
2015.03.15 23:52:35 -  INFO [RetrieveURL.java:153]: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM257244.zip
2015.03.15 23:52:44 -  INFO [RetrieveURL.java:100]: | | February 2015: Additions and Deletions to the Drug Product List (PDF - 127KB) | |
2015.03.15 23:52:44 -  INFO [RetrieveURL.java:101]: Title:: News & Events > What's New Related to Drugs
2015.03.15 23:52:44 -  INFO [RetrieveURL.java:102]: February 2015: Additions and Deletions to the Drug Product List (PDF - 127KB)
2015.03.15 23:52:44 -  INFO [RetrieveURL.java:103]: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM438105.pdf
```

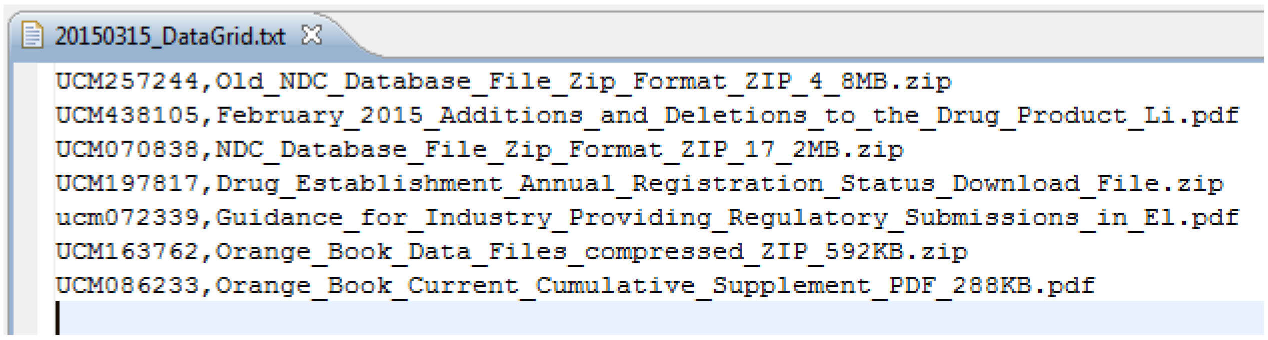


```
2015.03.15 23:52:44 -  INFO [RetrieveURL.java:100]: | |  | |
2015.03.15 23:52:44 -  INFO [RetrieveURL.java:101]: Title:: News & Events > What's New Related to Drugs
2015.03.15 23:52:44 -  INFO [RetrieveURL.java:102]:
2015.03.15 23:52:44 -  INFO [RetrieveURL.java:103]: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM086233.pdf
2015.03.15 23:52:45 -  INFO [RetrieveURL.java:100]: | | Orange Book Current Cumulative Supplement (PDF - 288KB) | |
2015.03.15 23:52:45 -  INFO [RetrieveURL.java:101]: Title:: News & Events > What's New Related to Drugs
2015.03.15 23:52:45 -  INFO [RetrieveURL.java:102]: Orange Book Current Cumulative Supplement (PDF - 288KB)
2015.03.15 23:52:45 -  INFO [RetrieveURL.java:103]: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM086233.pdf
2015.03.15 23:52:45 -  INFO [RetrieveURL.java:100]: | | Orange Book Data Files (compressed) (ZIP - 592KB) | |
2015.03.15 23:52:45 -  INFO [RetrieveURL.java:101]: Title:: News & Events > What's New Related to Drugs
2015.03.15 23:52:45 -  INFO [RetrieveURL.java:102]: Orange Book Data Files (compressed) (ZIP - 592KB)
2015.03.15 23:52:46 -  INFO [RetrieveURL.java:103]: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM163762.zip
2015.03.15 23:52:46 -  INFO [RetrieveURL.java:189]: Total No. of PDF or ZIP files found: 8
2015.03.15 23:52:46 -  INFO [RetrieveURL.java:196]: Path to save files:D:\Sri Kumaran Cabinet\BITS WILP\Fourth Semester\My Project\CODE\workspace_web_mining\Fi
2015.03.15 23:52:46 -  INFO [RetrieveURL.java:203]:  - 1 -
2015.03.15 23:52:46 -  INFO [Downloader.java:51]: Downloading file from: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM257244.zip
2015.03.15 23:52:47 -  INFO [Downloader.java:56]: HTTP/1.1 200 OK
2015.03.15 23:54:19 -  INFO [Downloader.java:76]: Downloded 5045206 bytes. Content-Type: application/zip
2015.03.15 23:54:19 -  INFO [RetrieveURL.java:203]:  - 2 -
2015.03.15 23:54:19 -  INFO [Downloader.java:51]: Downloading file from: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM438105.pdf
2015.03.15 23:54:20 -  INFO [Downloader.java:56]: HTTP/1.1 200 OK
2015.03.15 23:54:22 -  INFO [Downloader.java:76]: Downloded 130154 bytes. Content-Type: Application/pdf
2015.03.15 23:54:22 -  INFO [RetrieveURL.java:203]:  - 3 -
2015.03.15 23:54:22 -  INFO [Downloader.java:51]: Downloading file from: http://www.fda.gov/downloads/Drugs/DevelopmentApprovalProcess/UCM070838.zip
2015.03.15 23:54:24 -  INFO [Downloader.java:56]: HTTP/1.1 200 OK
2015.03.16 00:02:11 -  INFO [Downloader.java:76]: Downloded 18040296 bytes. Content-Type: application/zip
2015.03.16 00:02:11 -  INFO [RetrieveURL.java:203]:  - 4 -
2015.03.16 00:02:11 -  INFO [Downloader.java:51]: Downloading file from: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM197817.zip
2015.03.16 00:02:13 -  INFO [Downloader.java:56]: HTTP/1.1 200 OK
2015.03.16 00:02:23 -  INFO [Downloader.java:76]: Downloded 375566 bytes. Content-Type: application/zip
2015.03.16 00:02:23 -  INFO [RetrieveURL.java:203]:  - 5 -
2015.03.16 00:02:23 -  INFO [Downloader.java:51]: Downloading file from: http://www.fda.gov/downloads/Drugs/GuidanceComplianceRegulatoryInformation/Guidances/u
2015.03.16 00:02:24 -  INFO [Downloader.java:56]: HTTP/1.1 200 OK
2015.03.16 00:02:27 -  INFO [Downloader.java:76]: Downloded 105812 bytes. Content-Type: Application/pdf
```



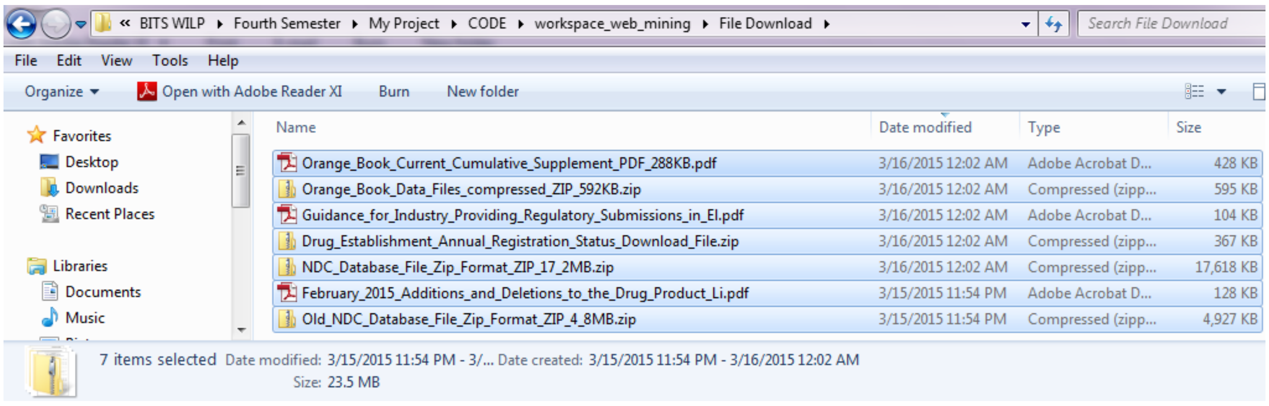
```
2015.03.16 00:02:27 -  INFO [RetrieveURL.java:203]:  - 6 -
2015.03.16 00:02:27 -  INFO [Downloader.java:51]: Downloading file from: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM163762.zip
2015.03.16 00:02:28 -  INFO [Downloader.java:56]: HTTP/1.1 200 OK
2015.03.16 00:02:42 -  INFO [Downloader.java:76]: Downloded 608524 bytes. Content-Type: application/zip
2015.03.16 00:02:42 -  INFO [RetrieveURL.java:203]:  - 7 -
2015.03.16 00:02:42 -  INFO [Downloader.java:51]: Downloading file from: http://www.fda.gov/downloads/Drugs/InformationOnDrugs/UCM086233.pdf
2015.03.16 00:02:43 -  INFO [Downloader.java:56]: HTTP/1.1 200 OK
2015.03.16 00:02:54 -  INFO [Downloader.java:76]: Downloded 438246 bytes. Content-Type: Application/pdf
2015.03.16 00:02:54 -  WARN [RetrieveURL.java:240]: DUPLICATE ENTRIES has been EXCLUDED by hash map
2015.03.16 00:02:54 -  INFO [RetrieveURL.java:241]: Total No. of PDF or ZIP files downloaded: 7
2015.03.16 00:02:54 -  INFO [MainET.java:85]: Execution started on: Sun Mar 15 23:51:22 IST 2015
2015.03.16 00:02:54 -  INFO [MainET.java:88]: Execution ended on: Mon Mar 16 00:02:54 IST 2015
2015.03.16 00:02:56 -  INFO [MainET.java:102]: --COMPLETED--
```

20150315_DataGrid.txt



```
UCM257244,Old_NDC_Database_File_Zip_Format_ZIP_4_8MB.zip
UCM438105,February_2015_Additions_and_Deletions_to_the_Drug_Product_Li.pdf
UCM070838,NDC_Database_File_Zip_Format_ZIP_17_2MB.zip
UCM197817,Drug_Establishment_Annual_Registration_Status_Download_File.zip
ucm072339,Guidance_for_Industry_Providing_Regulatory_Submissions_in_El.pdf
UCM163762,Orange_Book_Data_Files_compressed_ZIP_592KB.zip
UCM086233,Orange_Book_Current_Cumulative_Supplement_PDF_288KB.pdf
```

D:\Sri Kumaran Cabinet\BITS WILP\Fourth Semester\My Project\CODE\workspace_web_mining\File Download



« BITS WILP ▸ Fourth Semester ▸ My Project ▸ CODE ▸ workspace_web_mining ▸ File Download ▸
Search File Download
File Edit View Tools Help
Organize Open with Adobe Reader XI Burn New folder
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music

Name	Date modified	Type	Size
Orange_Book_Current_Cumulative_Supplement_PDF_288KB.pdf	3/16/2015 12:02 AM	Adobe Acrobat D...	428 KB
Orange_Book_Data_Files_compressed_ZIP_592KB.zip	3/16/2015 12:02 AM	Compressed (zipp...	595 KB
Guidance_for_Industry_Providing_Regulatory_Submissions_in_El.pdf	3/16/2015 12:02 AM	Adobe Acrobat D...	104 KB
Drug_Establishment_Annual_Registration_Status_Download_File.zip	3/16/2015 12:02 AM	Compressed (zipp...	367 KB
NDC_Database_File_Zip_Format_ZIP_17_2MB.zip	3/16/2015 12:02 AM	Compressed (zipp...	17,618 KB
February_2015_Additions_and_Deletions_to_the_Drug_Product_Li.pdf	3/15/2015 11:54 PM	Adobe Acrobat D...	128 KB
Old_NDC_Database_File_Zip_Format_ZIP_4_8MB.zip	3/15/2015 11:54 PM	Compressed (zipp...	4,927 KB

7 items selected Date modified: 3/15/2015 11:54 PM - 3/... Date created: 3/15/2015 11:54 PM - 3/16/2015 12:02 AM
Size: 23.5 MB

Chapter 3: Loading (L)

3.1 Overview

Extracted and finely refined contents are very precious, vital and will be used in most critical phases of Pharma lifecycle. So it's time to save Extracted content to a secure location. In our project, secure location is Content Management System. It is basically Global Electronic Pharmaceutical Information Center. Extracted content and FDA Regulatory content are shared by same Content Management System, this system should be completely validated and comply with FDA standards – International Conference on Harmonisation's 21 Code of Federal Regulation and Good Automated Manufacturing Practice.

3.2 Introduction to Content Management Systems in Life Science sector

A Content Management System (CMS) is a computer application that allows publishing, editing and modifying content, organizing, deleting as well as maintenance from a central interface. Such systems of content management provide procedures to manage workflow in a collaborative environment. These procedures can be manual steps or an automated cascade. In Life Science, CMS is used for authoring, reviewing, approving and storing submission and submission supporting documentation which include extracted content from FDA Website. CMS is used by R&D functional groups, i.e., Research, Clinical Research, Regulatory, Clinical Biometrics, Safety, Global Quality, Non-clinical, CMC. CMS provides version control and standardized electronic formats.

3.3 Why Documentum?

Documentum is Unified Platform, Uncompromised Compliance, Seamless Content Control, Flexible and Trusted Cloud. Documentum provides dedicated modules for Life Science R&D life cycle. Modules: Electronic Trail Master File, Research and Development, Submission Store and View, Quality and Manufacturing.

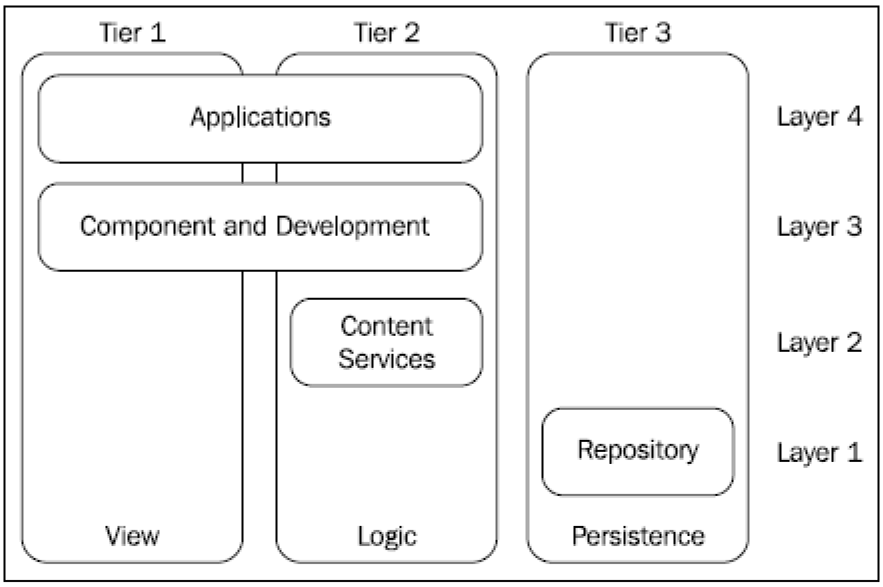


Figure 3-1: Documentum Architecture

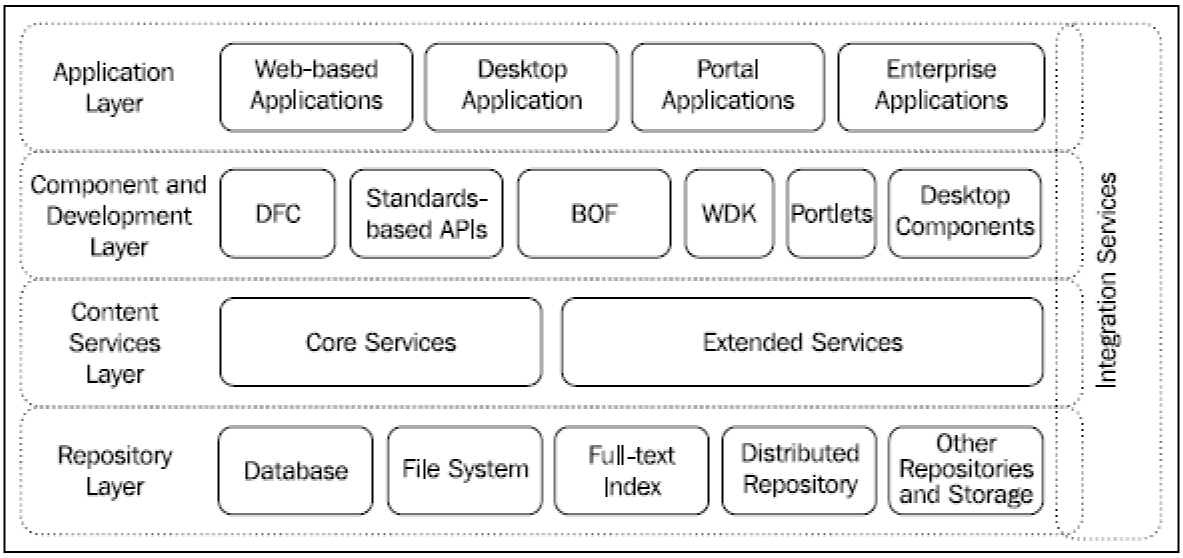


Figure 3-2: Documentum Layers

The Repository Layer provides storage for the platform and consists of the content repository, which uses file stores and a relational database as its components.

The Content Services Layer provides application-level services for organizing, controlling, sequencing, and delivering content to and from the repository.

The Component and Development Layer provides access to the repository content and the content services. This layer consists of predefined components and their application programming interfaces for enabling customization, integration, and application development. This layer consists of Documentum Foundation Classes (DFC), a set of standards-based APIs, Business Object Framework, Webtop Development Kit (WDK), Portlets, and Desktop components. The Component and Development Layer builds the bridge to the content services layer for applications that are part of the Application Layer. It is the Application Layer that makes the platform available to human users.

The application layer essentially opens up the platform for any type of use that can utilize content-management capabilities. The Applications in this Layer can be categorized into web-based applications, desktop applications, portal applications, and enterprise applications.

3.4 Documentum Foundation Classes and Documentum Query Language

DFC is Documentum Foundation Classes, a key part of the Documentum software platform. While the main user of DFC is other Documentum software, we can use DFC in any of the following ways:
• Access Documentum functionality from within one of enterprise applications.
For example, a publishing application can retrieve a submission document from Documentum system.
• Customize or extend products like Documentum Desktop or Webtop.
For example, we can modify Webtop functionality to implement one of Pharma's business rules.
• Write a method or procedure for Content Server to execute as part of a workflow or document lifecycle.
For example, the procedure that runs when we promote an XML document might apply a transformation to it and start a workflow to subject the transformed document to a predefined business process.

Table 3-1: Documentum Layers and Tiers

Repositories	One or more places where we keep the content and associated metadata of our organization's information. The metadata resides in a relational database, and the content resides in various storage elements.
Content Server	Software that manages, protects, and imposes an object oriented structure on the information in repositories. It provides intrinsic tools for managing the lifecycles of that information and automating processes for manipulating it.
Client programs	Software that provides interfaces between Content Server and end users. The most common clients run on application servers *(for example, Webtop)* or on personal computers *(for example, Desktop).*
End Users	People who control, contribute, or use our organization's information. They use a browser to access client programs running on application servers, or they use the integral user interface of a client program running on their personal computer.

In this view of Documentum functionality, Documentum Foundation Classes (DFC) lies between Content Server and clients. Documentum Foundation Services are the primary client interface to the Documentum platform. Documentum Foundation Classes are used for server‑side business logic and customization. DFC is Java based. As a result, client programs that are Java based can interface directly with DFC.

Where is DFC?

DFC runs on a Java virtual machine (JVM), which can be on:

• The machine that runs Content Server.

For example, to be called from a method as part of a workflow or document lifecycle.

• A middle‑tier system.

For example, on an application server to support WDK or to execute server methods.

DFC Packages:

DFC comprises a number of packages, that is, sets of related classes and interfaces.

• The names of DFC Java classes begin with Df *(for example, DfCollectionX).*

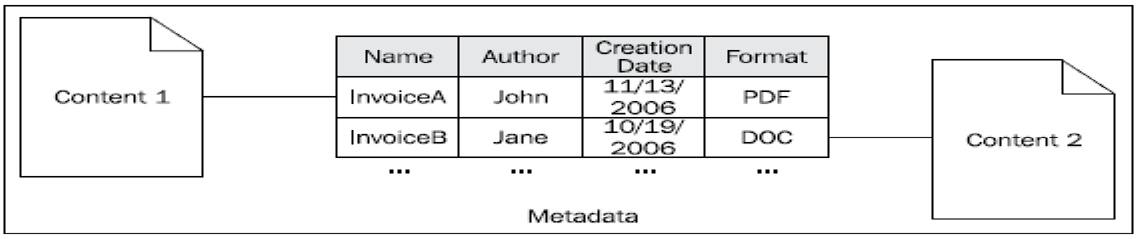
• Names of interfaces begin with IDf *(for example, IDfSessionManager).*

Interfaces expose DFC's public methods. Each interface contains a set of related methods.

Object-oriented Model:

The Content Server uses an object-oriented model and stores everything as an object in the repository.

For Example: Folder, Document. Document Content, Meta data.



Name	Author	Creation Date	Format
InvoiceA	John	11/13/2006	PDF
InvoiceB	Jane	10/19/2006	DOC
...	...	...	...

Figure 3-3: Content and Meta data relationship

Object Type:

An object type is a template for creating objects. In other words, an object is an instance of its type. A Documentum repository contains many predefined types and allows addition of new user-defined types (also known as custom types).

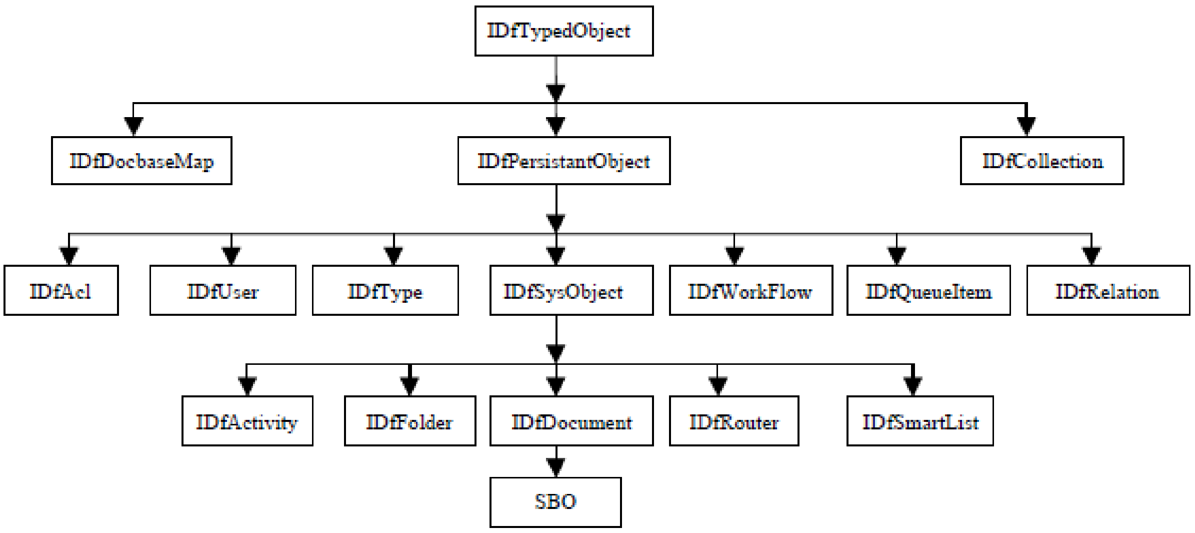


Figure 3-4: Documentum Object Hierarchy

DQL:

Metadata can be retrieved using Document Query Language (DQL), which is a superset of Structured Query Language used with databases (ANSI SQL). DQL can query objects and their relationships, in addition to any database tables registered to be queried using DQL.

Data dictionary:

Data dictionary stores information about object types in the repository. The default data dictionary can be altered by addition of user-defined object types and properties. Properties are also known as attributes and the two terms are used interchangeably to refer to metadata.

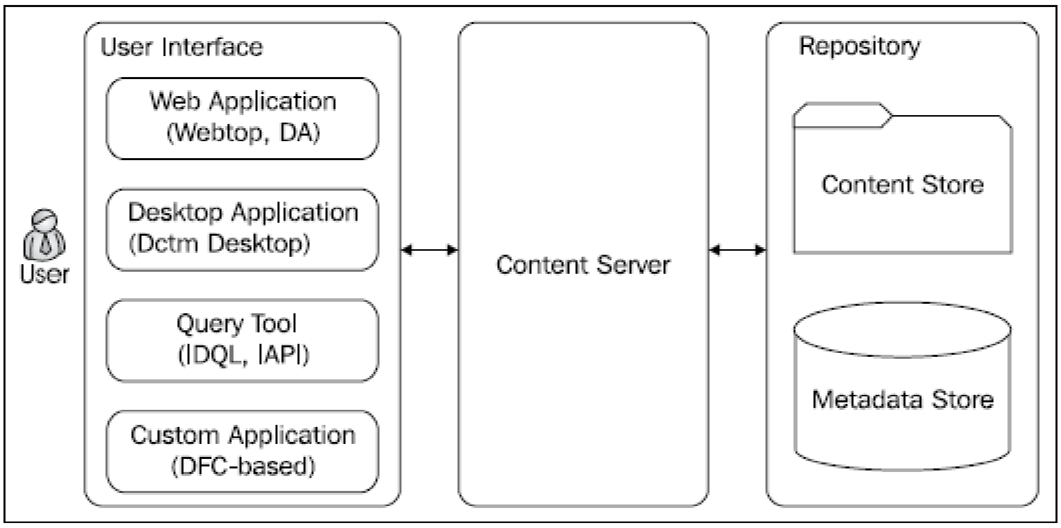

User
User Interface
Web Application (Webtop, DA)
Desktop Application (Dctm Desktop)
Query Tool (IDQL, IAPI)
Custom Application (DFC-based)
Content Server
Repository
Content Store
Metadata Store

Figure 3-5: Documentum – A Simple View

In simple terms, DFC is Object-oriented Documentum built Java based classes, which implicitly executes DQL for all the types of manipulations of bulk loads. *For Example: DQL UPDATE query can accommodate very few. DFC can accommodate unlimited entries with the help of data structures like Hash Map.*

3.5 Documentum in our Project Scope

In FDA Web Content Mining Project, the contents and it details are Extracted & Transformed from FDA Site and Loaded into Documentum. Actually, the content which are available at FDA Site was published with the help of Documentum Web Publisher.

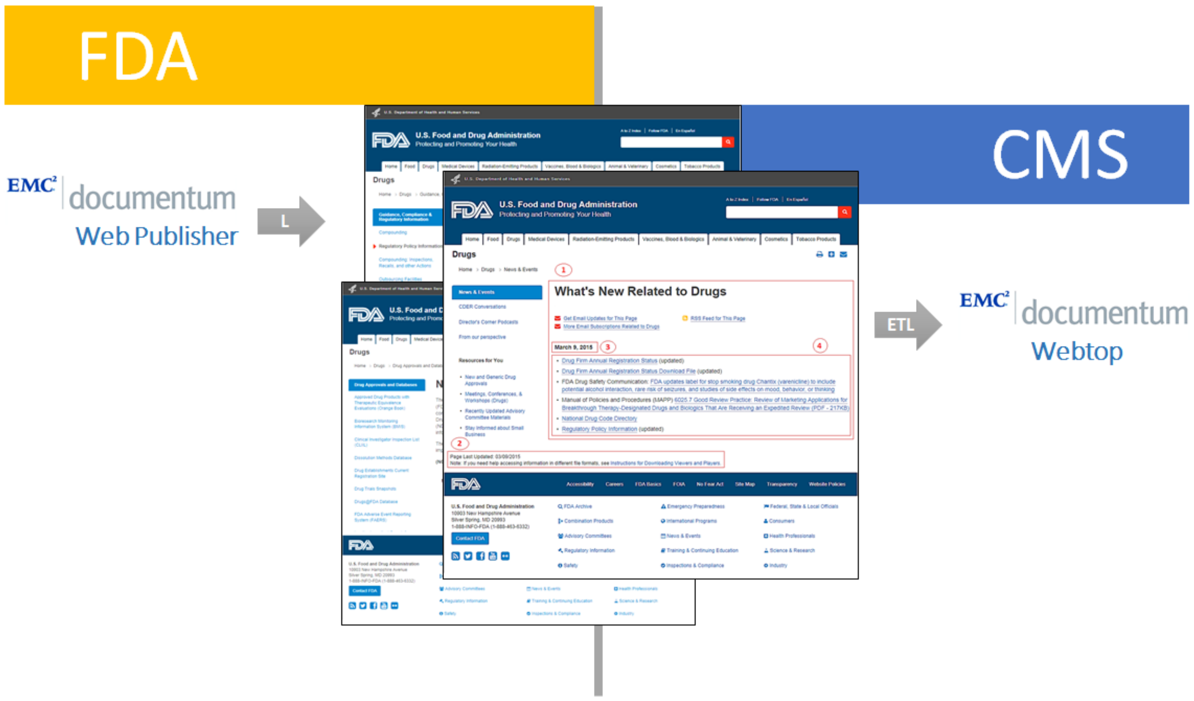

FDA
CMS
EMC² documentum
Web Publisher
L
ETL
EMC² documentum
Webtop
U.S. Food and Drug Administration
Protecting and Promoting Your Health
Drugs
What's New Related to Drugs

Figure 3-6: FDA uses Documentum Web Publisher

Documentum Web Publisher:

Web Publisher is a browser-based application that simplifies and automates the creation, review, and publication of web content. It works within Documentum 5, using Documentum Content Server to store and process content. It uses Documentum Site Caching Services (SCS) to publish content to web. Web Publisher manages web content through its entire life: creation, review, approval, publishing and archiving. Web Publisher also includes a full complement of capabilities for global site management, faster web development using templates and presentation files, administration. Web Publisher can be integrated with authoring tools to develop web sites.

In Loading, FDA extracted contents and it details goes to R&D Module and FDA Web Content Mining Project & it Validation documents goes to Quality Module of Documentum Life Sciences Solution Suite.

Based on the Webpage information, the documents will be classified to cabinets/folders & document unit attribute of dm_document type will be set and loaded with the help of Import/Check-in functionalities.

3.6 Techniques used for Loading

Execution Flow:

1. Set-up Session between client and server with the help of Documentum Session Manager – DFC and components.
2. Collect list of document details from text files (File System) to a collection. This text file has the information about extracted document details. *For Example: {Title}, {Name}.*
3. Collect list of document details from Documentum Docbase to a collection.
4. Compare both the collections and load matched & unmatched documents in separate collections.
5. Matched documents will be versioned with the existing document at Docbase.

6. Unmatched documents will be imported as new objects at Docbase.
7. Close all sessions.

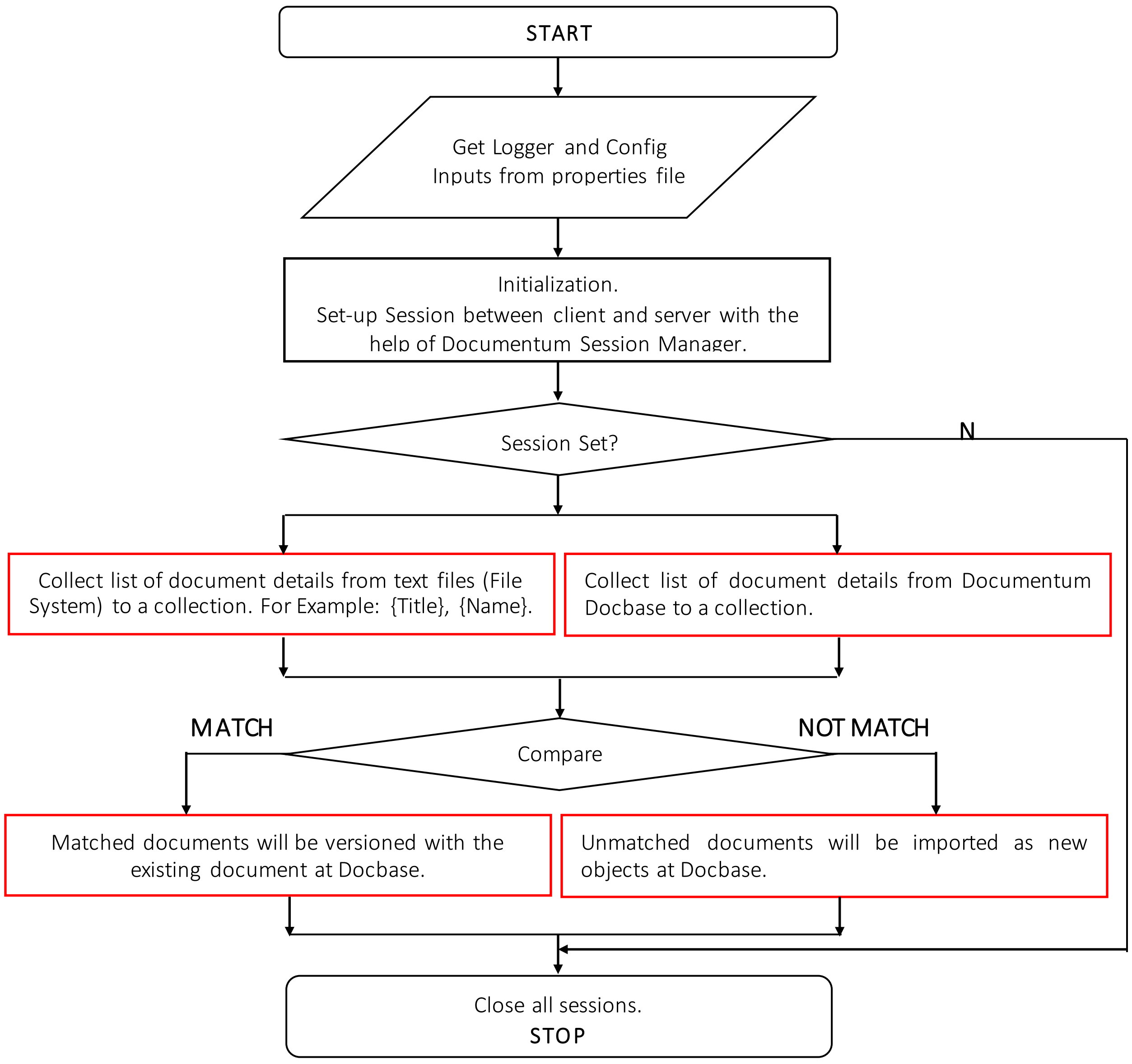

START
Get Logger and Config Inputs from properties file
Initialization. Set-up Session between client and server with the help of Documentum Session Manager.
Session Set?
N
Collect list of document details from text files (File System) to a collection. For Example: {Title}, {Name}.
Collect list of document details from Documentum Docbase to a collection.
MATCH
Compare
NOT MATCH
Matched documents will be versioned with the existing document at Docbase.
Unmatched documents will be imported as new objects at Docbase.
Close all sessions.
STOP

Figure 3-7: Loading – Execution Flow

Session setup:

Documentum architecture follows the client/server model. DFC-based applications are client programs even if DFC runs on the same machine as Content Server. The getLocalClient() method of DfClientX initializes DFC. The instantiation of DFC using the getLocalClient() method performs much of the runtime context initialization related tasks such as assigning policy permissions, loading the DFC property file, creating the DFC identity on the global repository, and registering the DFC identity in the global repository. The getLocalClient() method returns an object of IDfClient type that provides methods to retrieve Repository, Docbroker, and Server information. In addition, the object provides methods to create session manager. IDfSessionManager is to get/release sessions.

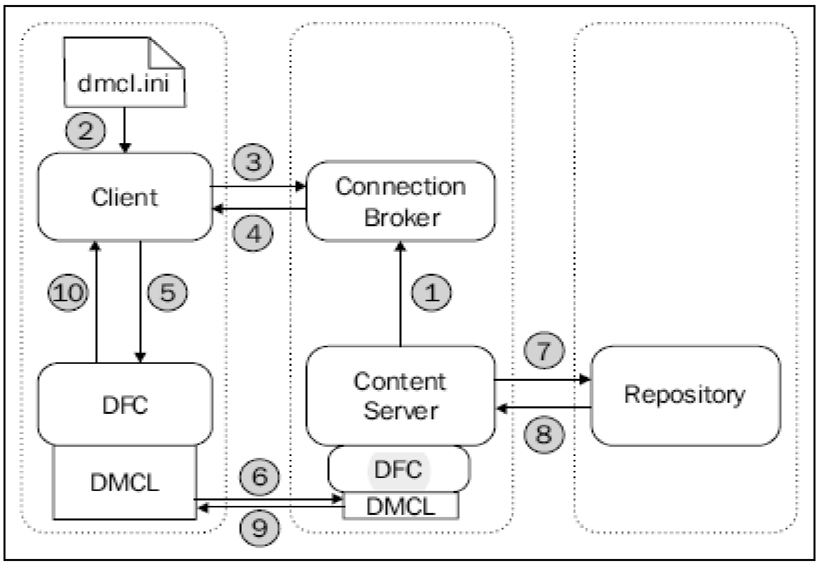


Figure 3-8: Documentum – Fundamental Communication Pattern

Analysis on Java Collections:

Collection	Description/features
Set	To store non duplicate elements
List	Can store duplicate elements
Map	To store elements with accessing keys(indexes)

Figure 3-9: Java Collection Framework (JCF)

Table 3-2: A quick comparison on Java Collections

	ArrayList	Vector	LinkedList	HashMap	LinkedHashMap	HashTable	TreeMap	HashSet	LinkedHashSet	TreeSet
Allows Null?	Yes	Yes	Yes	Yes (But One Key & Multiple Values)	Yes (But One Key & Multiple Values)	No	Yes (But Zero Key & Multiple Values)	Yes	Yes	No
Allows Duplicates?	Yes	Yes	Yes	No	No	No	No	No	No	No
Retrieves Sorted Results?	No	No	No	No	No	No	Yes	No	No	Yes
Retrieves Same as Insertion Order?	Yes	Yes	Yes	No	Yes	No	No	No	Yes	No
Synchronized?	No	Yes	No	No	No	Yes	No	No	No	No

Table 3-3: Big-O forList, Set, Map

Structure	get	add	remove	contains
ArrayList	O(1)	O(1)	O(n)	O(n)
LinkedList	O(n)	O(1)	O(1)	O(n)
HashSet	O(1)	O(1)	O(1)	O(1)
LinkedHashSet	O(1)	O(1)	O(1)	O(1)
TreeSet	O(log n)	O(log n)	O(log n)	O(log n)

Structure	get	put	remove	containsKey
HashMap	O(1)	O(1)	O(1)	O(1)
LinkedHashMap	O(1)	O(1)	O(1)	O(1)
TreeMap	O(log n)	O(log n)	O(log n)	O(log n)

Lists and Maps are different data structures. Maps are used for when we want to associate a key with a value and Lists are an ordered collection.

Map is an interface in the Java Collection Framework and a HashMap is one implementation of the Map interface. HashMap are efficient for locating a value based on a key and inserting and deleting values based on a key. The entries of a HashMap are not ordered.

ArrayList and LinkedList are an implementation of the List interface. LinkedList provides sequential access and is generally more efficient at inserting and deleting elements in the list, however, it is it less efficient at accessing elements in a list. ArrayList provides random access and is more efficient at accessing elements but is generally slower at inserting and deleting elements.

Why Hash Map is chosen?

Mainly, we need to collect Document Number as key and Document Name as value in Hash Map. We can store more than one values as a single value with the help of comma separation. Parsing of comma separated values are very low cost. Hash Map will not allow duplicate keys and lead a way to very fast data access/retrieval. Sorting is not needed, which improves the efficiency.

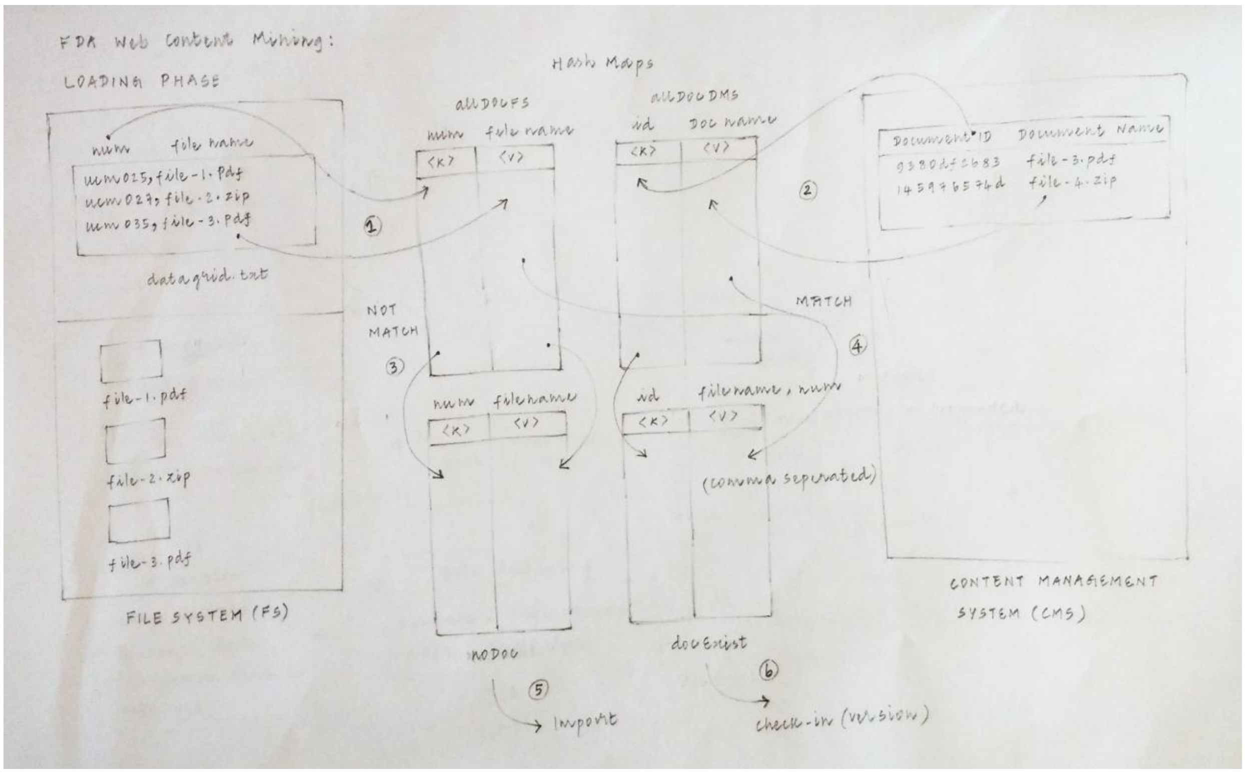


Figure 3-10: Loading – Technical Flow

3.7 Design

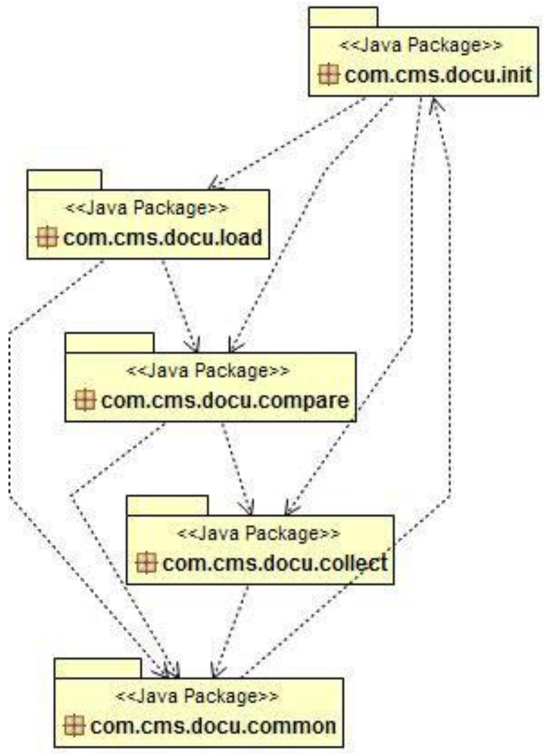
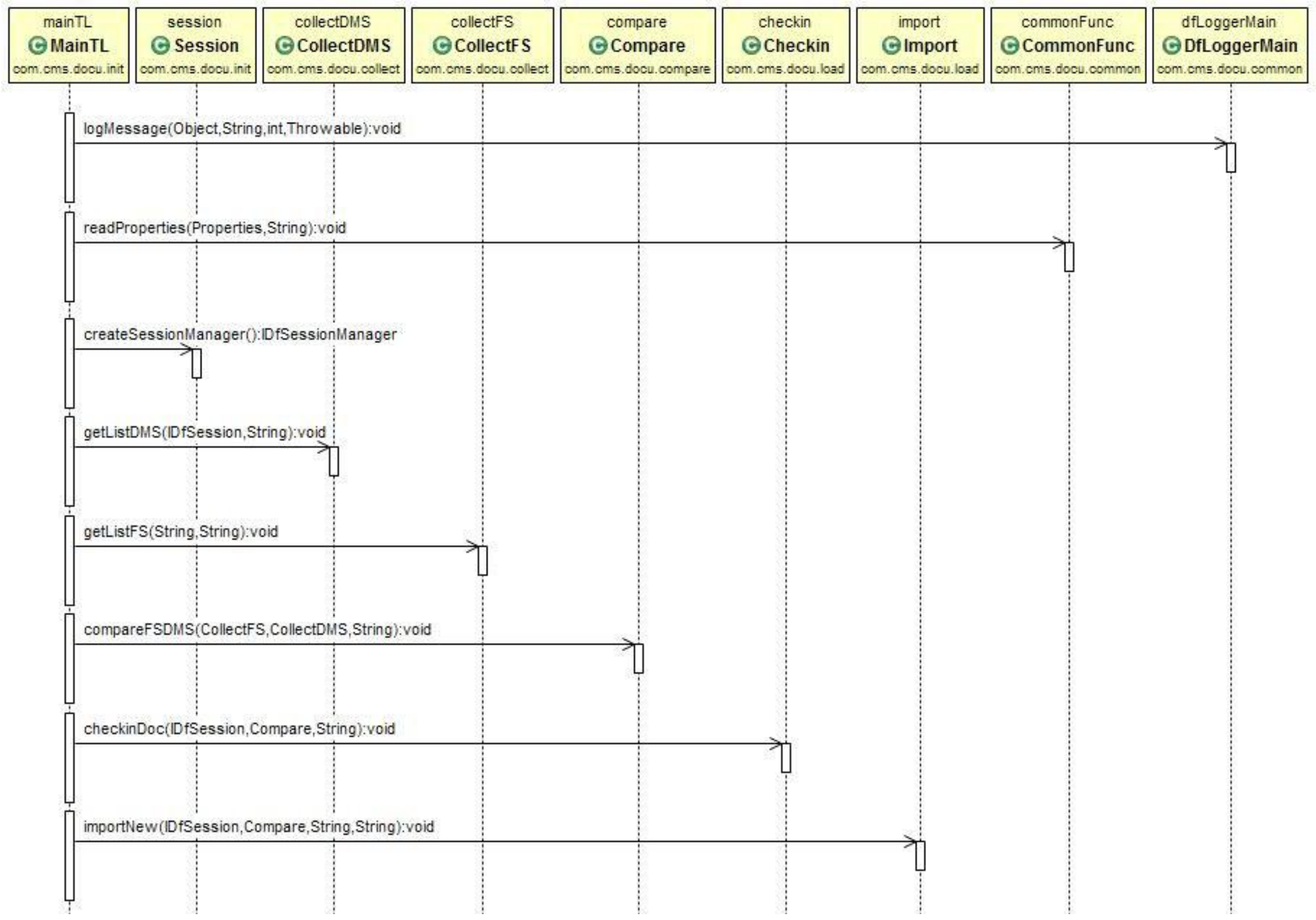
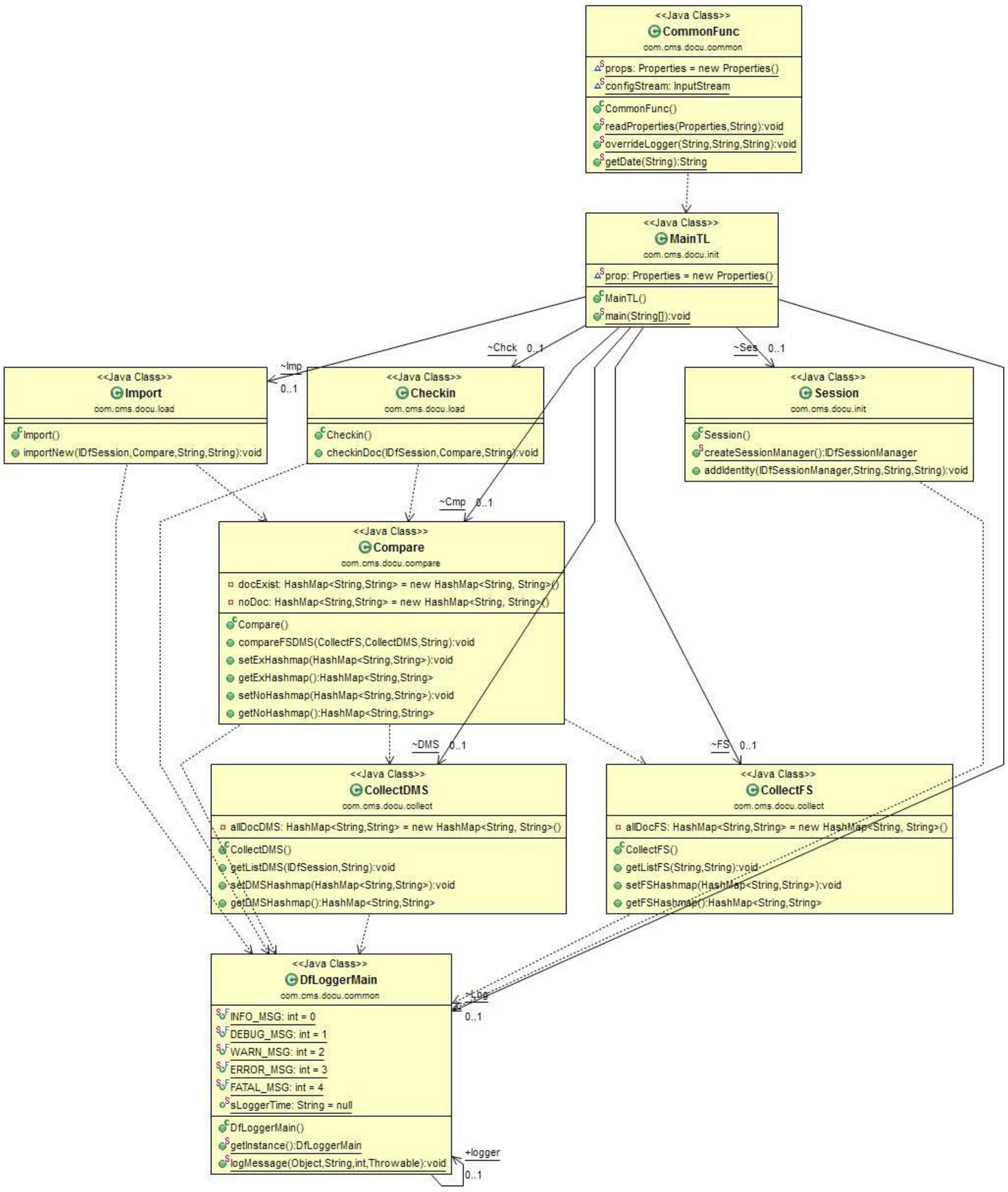


Figure 3-11: Loading – Package Diagram



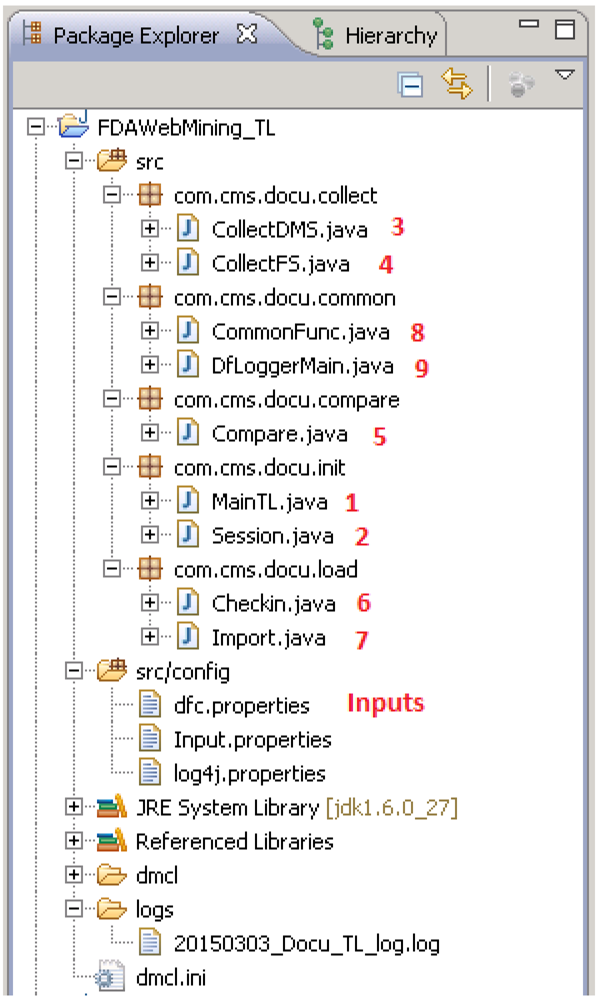
mainTL
MainTL
com.cms.docu.init
session
Session
com.cms.docu.init
collectDMS
CollectDMS
com.cms.docu.collect
collectFS
CollectFS
com.cms.docu.collect
compare
Compare
com.cms.docu.compare
checkin
Checkin
com.cms.docu.load
import
Import
com.cms.docu.load
commonFunc
CommonFunc
com.cms.docu.common
dfLoggerMain
DfLoggerMain
com.cms.docu.common
logMessage(Object,String,int,Throwable):void
readProperties(Properties,String):void
createSessionManager():IDfSessionManager
getListDMS(IDfSession,String):void
getListFS(String,String):void
compareFSDMS(CollectFS,CollectDMS,String):void
checkinDoc(IDfSession,Compare,String):void
importNew(IDfSession,Compare,String,String):void

Figure 3-12: Loading – Sequence Diagram



<<Java Class>>
CommonFunc
com.cms.docu.common
props: Properties = new Properties()
configStream: InputStream
CommonFunc()
readProperties(Properties,String):void
overrideLogger(String,String,String):void
getDate(String):String
<<Java Class>>
MainTL
com.cms.docu.init
prop: Properties = new Properties()
MainTL()
main(String[]):void
~Imp
0..1
~Chck
0..1
~Ses
0..1
<<Java Class>>
Import
com.cms.docu.load
Import()
importNew(IDfSession,Compare,String,String):void
<<Java Class>>
Checkin
com.cms.docu.load
Checkin()
checkinDoc(IDfSession,Compare,String):void
<<Java Class>>
Session
com.cms.docu.init
Session()
createSessionManager():IDfSessionManager
addIdentity(IDfSessionManager,String,String,String):void
~Cmp
0..1
<<Java Class>>
Compare
com.cms.docu.compare
docExist: HashMap<String,String> = new HashMap<String, String>()
noDoc: HashMap<String,String> = new HashMap<String, String>()
Compare()
compareFSDMS(CollectFS,CollectDMS,String):void
setExHashmap(HashMap<String,String>):void
getExHashmap():HashMap<String,String>
setNoHashmap(HashMap<String,String>):void
getNoHashmap():HashMap<String,String>
~DMS
0..1
~FS
0..1
<<Java Class>>
CollectDMS
com.cms.docu.collect
allDocDMS: HashMap<String,String> = new HashMap<String, String>()
CollectDMS()
getListDMS(IDfSession,String):void
setDMSHashmap(HashMap<String,String>):void
getDMSHashmap():HashMap<String,String>
<<Java Class>>
CollectFS
com.cms.docu.collect
allDocFS: HashMap<String,String> = new HashMap<String, String>()
CollectFS()
getListFS(String,String):void
setFSHashmap(HashMap<String,String>):void
getFSHashmap():HashMap<String,String>
~Log
0..1
<<Java Class>>
DfLoggerMain
com.cms.docu.common
INFO_MSG: int = 0
DEBUG_MSG: int = 1
WARN_MSG: int = 2
ERROR_MSG: int = 3
FATAL_MSG: int = 4
sLoggerTime: String = null
DfLoggerMain()
getInstance():DfLoggerMain
logMessage(Object,String,int,Throwable):void
+logger
0..1

Figure 3-13: Loading – Class Diagram



```
FDAWebMining_TL
  src
    com.cms.docu.collect
      CollectDMS.java   3
      CollectFS.java   4
    com.cms.docu.common
      CommonFunc.java   8
      DfLoggerMain.java   9
    com.cms.docu.compare
      Compare.java   5
    com.cms.docu.init
      MainTL.java   1
      Session.java   2
    com.cms.docu.load
      Checkin.java   6
      Import.java   7
  src/config
    dfc.properties   Inputs
    Input.properties
    log4j.properties
  JRE System Library [jdk1.6.0_27]
  Referenced Libraries
  dmcl
  logs
    20150303_Docu_TL_log.log
  dmcl.ini
```

Figure 3-14: Loading – Framework

3.8 Code

```
package com.cms.docu.init;

import java.util.Properties;

import com.cms.docu.collect.CollectDMS;
import com.cms.docu.collect.CollectFS;
import com.cms.docu.common.DfLoggerMain;
import com.cms.docu.compare.Compare;
import com.cms.docu.load.Checkin;
import com.cms.docu.load.Import;
import com.documentum.fc.client.IDfSession;
import com.documentum.fc.client.IDfSessionManager;
import com.documentum.fc.common.DfException;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * MainTL: Create Session and call all the methods
 */
public class MainTL {
     static Properties   prop = new Properties();
     static DfLoggerMain Log  = new DfLoggerMain();
     static Session      Ses  = new Session();
     static CollectFS    FS   = new CollectFS();
     static CollectDMS   DMS  = new CollectDMS();
     static Compare      Cmp  = new Compare();
     static Import       Imp  = new Import();
     static Checkin      Chck = new Checkin();

     /**
      * This method is Main method. Read data from Properties file. Session will be created with
      * Documentum Docbase. Call all the methods of FDAWebMining_TL project.
      */
     public static void main(String args[]) {
            String propFileName = "Input.properties";
            String logPropFile = "log4j.properties";
            try {
                   com.cms.docu.common.CommonFunc.readProperties(prop, propFileName);
                   // Override Logger properties
```

```
                if (prop.getProperty("overrideLogProp").trim().toString().equals("true")) {
                    com.cms.docu.common.CommonFunc.overrideLogger(logPropFile, prop.getProperty("overrideAppender")
                            .trim(), "./logs/" + com.cms.docu.common.CommonFunc.getDate("yyyyMMdd") + "_"
                            + prop.getProperty("logFileName").trim());
                }
                // Create Session
                IDfSessionManager sessMgr = Session.createSessionManager();
                // Call Session Identifier with Docbase credentials and details
                Ses.addIdentity(sessMgr, prop.getProperty("username").trim(), prop.getProperty(
                        "password").trim(), prop.getProperty("repoName").trim());
                IDfSession sess = sessMgr.getSession(prop.getProperty("repoName").trim());
                // Get list of documents from Documentum Docbase
                DMS.getListDMS(sess, prop.getProperty("dmsPath").trim());
                // Get list of documents from List file at File System
                FS.getListFS(prop.getProperty("filePath").trim(), prop.getProperty("file").trim());
                // Compare File System & Docbase
                Cmp.compareFSDMS(FS, DMS, prop.getProperty("filePath").trim());
                // Import not matched documents
                Imp.importNew(sess, Cmp, prop.getProperty("filePath").trim(), prop.getProperty(
                        "destFldrId").trim());
                // Checkin matched documents
                Chck.checkinDoc(sess, Cmp, prop.getProperty("filePath").trim());
            } catch (DfException ex) {
                ex.printStackTrace();
                DfLoggerMain.logMessage(MainTL.class, "DfException: ", 3, ex);
            } catch (Exception ex) {
                ex.printStackTrace();
                DfLoggerMain.logMessage(MainTL.class, "Exception: ", 3, ex);
            } finally{
                DfLoggerMain.logMessage(MainTL.class, "--COMPLETED--", 0, null);
            }
        }
    }
```

```
package com.cms.docu.init;

import com.cms.docu.common.DfLoggerMain;
import com.documentum.com.DfClientX;
import com.documentum.com.IDfClientX;
```

```java
import com.documentum.fc.client.IDfClient;
import com.documentum.fc.client.IDfSessionManager;
import com.documentum.fc.common.DfException;
import com.documentum.fc.common.IDfLoginInfo;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * Session: Setup a Session and add identity to it
 */
public class Session {
     /**
      * This method is to setup Session Manager.
      */
     public static IDfSessionManager createSessionManager() throws DfException {
            IDfClientX clientX = new DfClientX();
            IDfClient localClient = clientX.getLocalClient();
            IDfSessionManager sessMgr = localClient.newSessionManager();
            return sessMgr;
     }

     /**
      * This method is to add identity to session which is under progress.
      *
      * @param sm
      *            contains the Session Manager object
      * @param username
      *            contains the username to login Docbase
      * @param password
      *            contains the password to login Docbase
      * @param repoName
      *            contains the Docbase name
      */
     public void addIdentity(IDfSessionManager sm, String username, String password, String repoName) {
            try {
                   IDfClientX clientX = new DfClientX();
                   IDfLoginInfo li = clientX.getLoginInfo();
                   li.setUser(username);
                   li.setPassword(password);
                   // Check whether session manager already has an identity
                   if (sm.hasIdentity(repoName)) {
                        sm.clearIdentity(repoName);
                  }
                  sm.setIdentity(repoName, li);
            } catch (DfException ex) {
                  ex.printStackTrace();
                  DfLoggerMain.logMessage(Session.class, "DfException: ", 3, ex);
            } catch (Exception ex) {
                  ex.printStackTrace();
                  DfLoggerMain.logMessage(Session.class, "Exception: ", 3, ex);
            }
      }
}
```

```
package com.cms.docu.collect;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Iterator;

import com.cms.docu.common.DfLoggerMain;
import com.documentum.fc.client.DfQuery;
import com.documentum.fc.client.IDfCollection;
import com.documentum.fc.client.IDfQuery;
import com.documentum.fc.client.IDfSession;
import com.documentum.fc.client.IDfSysObject;
import com.documentum.fc.common.DfException;
import com.documentum.fc.common.DfId;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * CollectDMS: Collect list of documents from Docbase
 */
public class CollectDMS {
     private HashMap<String, String> allDocDMS = new HashMap<String, String>();

     /**
      * This method is to collect list of documents from Docbase
      *
      * @param sess
      *            contains IDfSession object
      * @param dmsPath
      *            contains document path at Docbase
      */
     public void getListDMS(IDfSession sess, String dmsPath) throws Exception {
           IDfQuery query = null;
           IDfCollection colDQL = null;
           ArrayList<String> arrObjId = null;
           String sQuery = null;
           int ctr = 0;
           try {
                 arrObjId = new ArrayList<String>();
                 query = new DfQuery();
                 // DQL query to read object ids of latest version documents for a
                 // specified path
                 sQuery = "SELECT r_object_id FROM dm_document WHERE FOLDER('" + dmsPath
                         + "',descend) AND i_latest_flag = 1 enable(ROW_BASED)";
                 query.setDQL(sQuery);
                 colDQL = query.execute(sess, IDfQuery.DF_EXEC_QUERY);
                 if (colDQL != null) {
                       while (colDQL.next()) {
                             // Array to collect object ids
                             arrObjId.add((colDQL.getString("r_object_id").trim()));
                       }
                 }
                 colDQL.close();
                 if (arrObjId.size() > 1) {
                       DfLoggerMain.logMessage(CollectDMS.class, "No. of Documents in Docbase: "
                               + arrObjId.size(), 0, null);
                       // Iterator to transfer object ids from Array
                       Iterator<String> itrObj = arrObjId.iterator();
                       while (itrObj.hasNext()) {
                             String Object_Id = (String) itrObj.next();
                             IDfSysObject Object = (IDfSysObject) sess.getObject(new DfId(Object_Id));
                             String Object_Name = Object.getObjectName();
                             // Store collected object ids and it object name to Hash Map
                             // (here object refers to document)
                             allDocDMS.put(Object_Id, Object_Name);
                             ctr = ctr + 1;
                             DfLoggerMain.logMessage(CollectDMS.class, " -" + ctr + "- " + Object_Id + " "
                                        + Object_Name + " " + Object.getTitle(), 1, null);
                        }
                } else {
                        DfLoggerMain.logMessage(CollectDMS.class, "NO DOCUMENTS found under " + dmsPath + " at Docbase", 2,
                                        null);
                }
                setDMSHashmap(allDocDMS);
        } catch (DfException ex) {
                ex.printStackTrace();
                DfLoggerMain.logMessage(CollectDMS.class, "DfException: ", 3, ex);
        } catch (Exception ex) {
                ex.printStackTrace();
                DfLoggerMain.logMessage(CollectDMS.class, "Exception: ", 3, ex);
        }
}

/**
 * This method is SETTER for allDocDMS Hash Map
 *
 * @param allDocDMS
 *             contains list of documents from Docbase
 */
public void setDMSHashmap(HashMap<String, String> allDocDMS) {
        this.allDocDMS = allDocDMS;
}

/**
 * This method is GETTER for allDocDMS Hash Map
 */
public HashMap<String, String> getDMSHashmap() {
        return this.allDocDMS;
}
}
```

```
package com.cms.docu.collect;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.LineNumberReader;
```

```java
import java.util.HashMap;

import com.cms.docu.common.DfLoggerMain;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * CollectFS: Collect list of documents from File System
 */
public class CollectFS {
     private HashMap<String, String> allDocFS = new HashMap<String, String>();

     /**
      * This method is to collect list of documents from List File at File System
      *
      * @param filePath
      *             contains the file path of list file at file system
      * @param file
      *             contains the list file name
      */
     public void getListFS(String filePath, String file) {
           BufferedReader br;
           String line;
           int ctr = 0;
           try {
                 LineNumberReader lineNumberReader = new LineNumberReader(
                         new FileReader(filePath + file));
                 lineNumberReader.skip(Long.MAX_VALUE);
                 DfLoggerMain.logMessage(CollectFS.class, "No. of Documents in File List at File System: "
                         + (lineNumberReader.getLineNumber() + 1), 0, null);
                 // Check if there are any data in file at File System
                 if (lineNumberReader.getLineNumber() > 1) {
                       // Read data from File System
                       br = new BufferedReader(new FileReader(filePath + file));
                       while ((line = br.readLine()) != null) {
                             String str[] = line.split(",");
                             // Store collected object titles and it object name in Hash Map
                             allDocFS.put(str[0].trim(), str[1].trim());
                             ctr = ctr + 1;
                             DfLoggerMain.logMessage(CollectFS.class, " -" + ctr + "- " + str[0] + " " + str[1], 1, null);
                       }
```

```
                        br.close();
                    } else {
                        DfLoggerMain.logMessage(CollectFS.class, "NO DOCUMENTS found under " + filePath
                                + file, 2, null);
                    }
                    setFSHashmap(allDocFS);
            } catch (Exception ex) {
                    ex.printStackTrace();
                    DfLoggerMain.logMessage(CollectFS.class, "Exception: ", 3, ex);
            }
    }

    /**
     * This method is SETTER for allDocFS Hash Map
     *
     * @param allDocFS
     *            contains list of documents from List File
     */
    public void setFSHashmap(HashMap<String, String> allDocFS) {
            this.allDocFS = allDocFS;
    }

    /**
     * This method is GETTER for allDocFS Hash Map
     */
    public HashMap<String, String> getFSHashmap() {
            return this.allDocFS;
    }
}
```

```
package com.cms.docu.compare;

import java.io.File;
import java.util.HashMap;
import java.util.Set;

import com.cms.docu.collect.CollectDMS;
import com.cms.docu.collect.CollectFS;
import com.cms.docu.common.DfLoggerMain;
```

```
/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * Compare: Compare documents between File System and Docbase
 */
public class Compare {
     private HashMap<String, String> docExist = new HashMap<String, String>();
     private HashMap<String, String> noDoc    = new HashMap<String, String>();

     /**
      * This method is to compare allDocFS and allDocDMS Hash Maps for matched and not matched
      * documents
      *
      * @param cFS
      *            object for CollectFS class
      * @param cDMS
      *            object for CollectDMS class
      * @param filePath
      *            contains file path of documents at File System
      */
     public void compareFSDMS(CollectFS cFS, CollectDMS cDMS, String filePath) {
           // Access allDocFS Hash Map from CollectFS class
           HashMap<String, String> allDocFS = cFS.getFSHashmap();
           // Access allDocDMS Hash Map from CollectDMS class
           HashMap<String, String> allDocDMS = cDMS.getDMSHashmap();
           File fCO;
           int counter = 0;
           DfLoggerMain.logMessage(Compare.class, "Compare begins..", 0, null);
           Set<String> keysFS = allDocFS.keySet();
           Set<String> keysDMS = allDocDMS.keySet();
           try {
                 goback: for (String keyF : keysFS) {
                       String filepath = filePath + allDocFS.get(keyF);
                       fCO = new File(filepath);
                       // Check whether the documents listed at List File are exist at
                       // File System
                       if (fCO.exists()) {
                             //This condition check will help when the Docbase with no documents at specified folder path
                             if (allDocDMS.size() > 1) {
                                   for (String keyD : keysDMS) {
```

```
                                DfLoggerMain.logMessage(Compare.class, "FS::" + allDocFS.get(keyF)
                                        + " DMS::" + allDocDMS.get(keyD), 1, null);
                                // MATCH FOUND between allDocFS and allDocDMS Hash Map
                                if (allDocFS.get(keyF).toString().trim().equals(
                                        allDocDMS.get(keyD).toString().trim())) {
                                      DfLoggerMain.logMessage(Compare.class, "Compare MATCH: " + keyD
                                              + " " + allDocDMS.get(keyD).toString() + " $ " + keyF, 0,
                                              null);
                                      // Store object ids, and it object name & object
                                      // title in Hash Map
                                      // (object name and object title separated by $ and
                                      // stored as a single value)
                                      docExist.put(keyD, allDocDMS.get(keyD).toString() + "$" + keyF);
                                      DfLoggerMain
                                              .logMessage(
                                                      Compare.class,
                                                      "Entry "
                                                              + keyD
                                                              + " "
                                                              + allDocDMS.get(keyD)

                                                   + " has been removed from DMS hashmap to avoid duplicate compare",
                                                      1, null);
                                      allDocDMS.remove(keyD);
                                      counter = 0;
                                      continue goback;
                                }
                                // NO MATCH FOUND between allDocFS and allDocDMS Hash Map
                                if (!(allDocFS.get(keyF).toString().trim().equals(allDocDMS.get(keyD)
                                        .toString().trim()))) {
                                      counter = counter + 1;
                                }
                                if (counter == keysDMS.size()) {
                                      DfLoggerMain.logMessage(Compare.class, "Compare NOT MATCH: " + keyF
                                              + " " + allDocFS.get(keyF).toString(), 0, null);
                                      // Store object titles and it object name in Hash
                                      // Map
                                      noDoc.put(keyF, allDocFS.get(keyF).toString());
                                }
                          }
                    } else {
```

```
                                DfLoggerMain.logMessage(Compare.class, "Compare NOT MATCH: " + keyF + " "
                                        + allDocFS.get(keyF).toString(), 0, null);
                                // Store object titles and it object name in Hash
                                // Map
                                noDoc.put(keyF, allDocFS.get(keyF).toString());
                            }
                    } else {
                            DfLoggerMain.logMessage(Compare.class, filepath + " does NOT EXIST!", 2, null);
                    }
                }
        } catch (Exception ex) {
                ex.printStackTrace();
                DfLoggerMain.logMessage(Compare.class, "DfException: ", 3, ex);
        }
}

/**
 * This method is SETTER for docExist Hash Map
 *
 * @param docExist
 *            contains list of MATCHED documents
 */
public void setExHashmap(HashMap<String, String> docExist) {
        this.docExist = docExist;
}

/**
 * This method is GETTER for docExist Hash Map
 */
public HashMap<String, String> getExHashmap() {
        return docExist;
}

/**
 * This method is SETTER for noDoc Hash Map
 *
 * @param noDoc
 *            contains list of NOT MATCHED documents
 */
public void setNoHashmap(HashMap<String, String> noDoc) {
        this.noDoc = noDoc;
	}

	/**
	 * This method is GETTER for noDoc Hash Map
	 */
	public HashMap<String, String> getNoHashmap() {
		return noDoc;
	}
}
```

```
package com.cms.docu.load;

import java.io.File;
import java.util.HashMap;
import java.util.Set;

import com.cms.docu.common.DfLoggerMain;
import com.cms.docu.compare.Compare;
import com.documentum.fc.client.IDfDocument;
import com.documentum.fc.client.IDfSession;
import com.documentum.fc.client.IDfVersionPolicy;
import com.documentum.fc.common.DfException;
import com.documentum.fc.common.DfId;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * Checkin: Check-in matched documents to Docbase
 */
public class Checkin {
	/**
	 * This method is to check-in MATCHED documents to Docbase.
	 *
	 * @param sess
	 *            contains IDfSession object
	 * @param Cmp
	 *            object for Compare class
	 * @param filePath
	 *            contains file path of documents at File System
 */
public void checkinDoc(IDfSession sess, Compare Cmp, String filePath) {
      // Access docExist Hash Map from Compare class
      HashMap<String, String> docExist = Cmp.getExHashmap();
      File fC;
      Set<String> keysExist = docExist.keySet();
      try {
            for (String KeyE : keysExist) {
                  // Document Name
                  String str = docExist.get(KeyE).toString();
                  // Document Title
                  String Nam_Title[] = str.split("\\$");
                  // Document from File System
                  String filepath = filePath + Nam_Title[0].trim();
                  fC = new File(filepath);
                  if (fC.exists()) {
                        IDfDocument dfDoc = (IDfDocument) sess.getObject(new DfId(KeyE));
                        // Check-out the document
                        dfDoc.checkout();
                        // Check-in the document
                        dfDoc.setFile(filepath);
                        // Set document title
                        dfDoc.setTitle(Nam_Title[1].trim());
                        IDfVersionPolicy vp = dfDoc.getVersionPolicy();
                        // Check-in as next major version
                        dfDoc.checkin(false, vp.getNextMajorLabel() + ",CURRENT");
                        DfLoggerMain.logMessage(Checkin.class, Nam_Title[0] + " " + Nam_Title[1].trim()
                                + " has been CHECKIN successfully", 0, null);
                  } else {
                        DfLoggerMain.logMessage(Checkin.class, filepath + " does NOT EXIST!", 2, null);
                  }
            }
      } catch (DfException ex) {
            ex.printStackTrace();
            DfLoggerMain.logMessage(Checkin.class, "DfException: ", 3, ex);
      } catch (Exception ex) {
            ex.printStackTrace();
            DfLoggerMain.logMessage(Checkin.class, "Exception: ", 3, ex);
      }}}
```

```java
package com.cms.docu.load;

import java.io.File;
import java.util.HashMap;
import java.util.Set;

import com.cms.docu.common.DfLoggerMain;
import com.cms.docu.compare.Compare;
import com.documentum.com.DfClientX;
import com.documentum.com.IDfClientX;
import com.documentum.fc.client.IDfDocument;
import com.documentum.fc.client.IDfSession;
import com.documentum.fc.common.DfException;
import com.documentum.fc.common.DfId;
import com.documentum.fc.common.IDfId;
import com.documentum.fc.common.IDfList;
import com.documentum.operations.IDfFile;
import com.documentum.operations.IDfImportNode;
import com.documentum.operations.IDfImportOperation;
import com.documentum.operations.IDfOperationError;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * Import: Import NOT MATCHED documents to Docbase
 */
public class Import {
     /**
      * This method is to import NOT MATCHED documents to Docbase.
      *
      * @param sess
      *            contains IDfSession object
      * @param Cmp
      *            object for Compare class
      * @param filePath
      *            contains file path of documents at File System
      * @param destFldrId
      *            contains document path of documents at Docbase
      */
     public void importNew(IDfSession sess, Compare Cmp, String filePath, String destFldrId)
             throws DfException {
```

```java
IDfClientX clientX = new DfClientX();
IDfImportOperation impOper = clientX.getImportOperation();
IDfId destId = new DfId(destFldrId);
// Access noDoc Hash Map from Compare class
HashMap<String, String> noDoc = Cmp.getNoHashmap();
File fI;
Set<String> keysNo = noDoc.keySet();
try {
      for (String KeyN : keysNo) {
            String docpath = filePath + noDoc.get(KeyN);
            fI = new File(docpath);
            if (fI.exists()) {
                  IDfFile localFile = clientX.getFile(docpath);
                  IDfImportNode impNode = (IDfImportNode) impOper.add(localFile);
                  impNode.setDestinationFolderId(destId);
                  // Object Type
                  impNode.setDocbaseObjectType("dm_document");
                  // Document Name
                  impNode.setNewObjectName(noDoc.get(KeyN).toString());
            } else {
                  DfLoggerMain.logMessage(Import.class, docpath + " does NOT EXIST!", 2, null);
            }
      }
      impOper.setSession(sess);
      if (impOper.execute()) {
            IDfList newObjLst = impOper.getNewObjects();
            int i = 0;
            goup: while (i < newObjLst.getCount()) {
                  for (String KeyN : keysNo) {
                        IDfDocument newObj = (IDfDocument) newObjLst.get(i);
                        if (noDoc.get(KeyN).equals(newObj.getObjectName())) {
                              // Document Title
                              newObj.setString("title", KeyN);
                              newObj.save();
                              DfLoggerMain.logMessage(Import.class, newObj.getObjectName() + " "
                                        + KeyN + " has been IMPORTED successfully", 0, null);
                              i++;
                              continue goup;
                        }
                  }
                  i++;
```

```
                        }
                } else {
                        // Import Error
                        IDfList errList = impOper.getErrors();
                        for (int i = 0; i < errList.getCount(); i++) {
                                IDfOperationError err = (IDfOperationError) errList.get(i);
                                DfLoggerMain.logMessage(Import.class, err.getMessage(), 3, null);
                        }
                }
        } catch (DfException ex) {
                ex.printStackTrace();
                DfLoggerMain.logMessage(Import.class, "DfException: ", 3, ex);
        } catch (Exception ex) {
                ex.printStackTrace();
                DfLoggerMain.logMessage(Import.class, "Exception: ", 3, ex);
        }
    }
}
```

```
package com.cms.docu.common;

import java.io.IOException;
import java.io.InputStream;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Properties;


import org.apache.log4j.LogManager;
import org.apache.log4j.PropertyConfigurator;

import com.cms.docu.init.MainTL;

/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * CommonFunc: Common Functions like read property file, override logger, format date
 */
```

```
public class CommonFunc {
     static Properties  props = new Properties();
     static InputStream configStream;

     /**
      * This method is to read properties from a specified properties file.
      *
      * @param prop
      *            contains properties object
      * @param propFile
      *            contains properties file name
      */
     public static void readProperties(Properties prop, String propFile) {
           try {
                 configStream = MainTL.class.getClassLoader().getResourceAsStream(propFile);
                 prop.load(configStream);
           } catch (IOException ex) {
                 ex.printStackTrace();
           }
     }

     /**
      * This method is to override Logger file properties.
      *
      * @param logPropFile
      *            contains logger properties file name
      * @param overrideAppender
      *            contains appender type, Ex.: File
      * @param logFileName
      *            contains logger file name
      */
     public static void overrideLogger(String logPropFile, String overrideAppender,
                 String logFileName) {
           try {
                 readProperties(props, logPropFile);
                 props.setProperty(overrideAppender, logFileName);
                 LogManager.resetConfiguration();
                 PropertyConfigurator.configure(props);
           } catch (Exception ex) {
                 ex.printStackTrace();
           }
```

```
	}

	/**
	 * This method is to Format date.
	 *
	 * @param format
	 *            contains format type, Ex.: yyyyMMdd
	 */
	public static String getDate(String format) throws ParseException {
		String nowDate = null;
		try {
			SimpleDateFormat sdfDate = new SimpleDateFormat(format);
			Date now = new Date();
			nowDate = sdfDate.format(now);
		} catch (Exception e) {
			e.printStackTrace();
		}
		return nowDate;
	}

}
```

```
package com.cms.docu.common;
import com.documentum.fc.common.DfLogger;
/* FDA Web Content Mining
 * @author :: Sri Kumaran Thiruppathy
 * @year :: 2015
 * DfLoggerMain: Setup custom DfLogger with levels 0-INFO, 1-DEBUG, 2-WARN, 3-ERROR, 4-FATAL
 */
public class DfLoggerMain {
	public static DfLoggerMain logger = null;
	public static final int INFO_MSG = 0;
	public static final int DEBUG_MSG = 1;
	public static final int WARN_MSG = 2;
	public static final int ERROR_MSG = 3;
	public static final int FATAL_MSG = 4;
	public static String sLoggerTime = null;
	/**
```

```
	 * constructor calling super class.
	 */
	public DfLoggerMain() {
		super();
	}
	/**
	 * This method is to create DfLoggerMain class instance.
	 */
	public static DfLoggerMain getInstance() {
		if (logger == null) {
			logger = new DfLoggerMain();
		}
		return logger;
	}

	/**
	 * This method is to log messages into the logger.
	 *
	 * @param source
	 *            contains the source class
	 * @param sMessage
	 *            contains the log message
	 * @param iTraceLevel
	 *            contains the level of logs
	 * @param ex
	 *            contains a throw object
	 */
	public static void logMessage(final Object source, final String sMessage,
			final int iTraceLevel, final Throwable ex) {
		if (iTraceLevel == INFO_MSG) {
			DfLogger.info(source, sMessage, null, ex);
		} else if (iTraceLevel == DEBUG_MSG) {
			DfLogger.debug(source, sMessage, null, ex);
		} else if (iTraceLevel == WARN_MSG) {
			DfLogger.warn(source, sMessage, null, ex);
		} else if (iTraceLevel == ERROR_MSG) {
			DfLogger.error(source, sMessage, null, ex);
		} else if (iTraceLevel == FATAL_MSG) {
			DfLogger.fatal(source, sMessage, null, ex);
		}
```

```
        }}
```

```
#dfc.properties#
dfc.data.dir=C\:/Documentum
dfc.docbroker.host[0]=j2epic02.mocr-nt1.otsuka.com
dfc.docbroker.port[0]=1489
dfc.globalregistry.repository=useretl
dfc.globalregistry.username=dm_bof_registry
dfc.globalregistry.password=cgvmkIbjXqxymVrNnybn/Q\=\=
dfc.registry.mode=file
```

```
#dmcl.ini#
[DOCBROKER_PRIMARY]
host = j2epic02
port = 1489
```

```
#Input.properties#
#Docbase Credentials#
username=useretl
password=useretl
repoName=epictest
#
#Destination Folder#
destFldrId=0b009c54803afaf4
#
#DMS Folder Path#
dmsPath=/Users/FDA
#
#File System File Path#
filePath=C:\\FDA\\files\\Scenario-2\\
#
#File System File List#
file=list_of_downloaded_file_details.txt
#
#Override Logger Properties
overrideLogProp=true
#Override File Appender
```

```
overrideAppender=log4j.appender.F1.File
#Override File name
 logFileName=Docu_TL_log.log
```

```
#log4j.properties#
log4j.rootCategory=DEBUG,A1,F1
log4j.category.MUTE=OFF
log4j.additivity.tracing=true
#log4j.category.tracing=DEBUG, FILE_TRACE
#-------------DEBUG<INFO<WARN<ERROR<FATAL--------------
#------------------- CONSOLE --------------------------
log4j.appender.A1=org.apache.log4j.ConsoleAppender
log4j.appender.A1.threshold=INFO
log4j.appender.A1.layout=org.apache.log4j.PatternLayout
log4j.appender.A1.layout.ConversionPattern=%d{yyyy.MM.dd HH:mm:ss} - %5p [%c{1}]: %m%n
#------------------- FILE -----------------------------
log4j.appender.F1=org.apache.log4j.FileAppender
log4j.appender.F1.File=./logs/log4j.log
log4j.appender.F1.threshold=DEBUG
log4j.appender.F1.append=true
log4j.appender.F1.layout=org.apache.log4j.PatternLayout
log4j.appender.F1.layout.ConversionPattern=%d{yyyy.MM.dd HH:mm:ss} - %5p [%c]: %m%n
#------------------- FILE_TRACE -----------------------
log4j.appender.FILE_TRACE=org.apache.log4j.RollingFileAppender
log4j.appender.FILE_TRACE.File=./logs/trace.log
log4j.appender.FILE_TRACE.MaxFileSize=100MB
log4j.appender.FILE_TRACE.layout=org.apache.log4j.PatternLayout
log4j.appender.FILE_TRACE.layout.ConversionPattern=%d{ABSOLUTE} [%t] %m%n
```

Chapter 4: Computer System Validation (CSV)

4.1 Introduction

Computer System Validation is the technical discipline that Life Science companies use to ensure that each Information Technology application fulfills its intended purpose. Stringent quality requirements in FDA regulated industries impose the need for specific controls and procedures throughout the Software Development Life Cycle (SDLC). Evidence that these controls and procedures have been followed and that they have resulted in quality software (software that satisfies its requirements) must be documented correctly and completely. These documents must be capable of standing up to close scrutiny by trained inspectors since the financial penalty for failing an audit can be extremely high. More importantly, a problem in a Life Science software application that affects the production environment could result in serious adverse consequences, including possible loss of life. The activities involved in applying the appropriate controls/procedures throughout the SDLC and for creating the necessary trail of documented evidence are all part of the technical discipline of Computer System Validation. As we will discuss in this article, software testing is a key component in this discipline. However, Computer System Validation, involves more than what many IT people consider to be software testing.

4.2 Is CSV a mixed fraction of IT Verification and Validation?

As applied to computer systems, the FDA definition of Validation is an umbrella term that is broader than the way the term *validation* is commonly used in the IT industry. In the IT industry, validation usually refers to *performing tests of software against its requirements*. A related term in the IT world is *verification*, which usually refers to *Inspections, Walkthroughs, and other reviews and activities aimed at ensuring that the results of successive steps* in the software development cycle correctly embrace the intentions of the previous step. As we will see below, FDA Validation of computer systems includes all of these activities with a key focus on producing documented evidence that will be readily available for inspection by the FDA. So testing in the sense of executing the software is only one of multiple techniques used in Computer System Validation.

4.3 Why CSV is extremely important in the Life Science sector?

1. Systematic Computer System Validation helps prevent software problems from reaching production environments. As previously mentioned, a problem in a Life Science software application that affects the production environment can result in serious adverse consequences. Besides the obvious humanistic reasons that the Life Science sector strives to prevent such harm to people, the business consequences of a software failure affecting people adversely can include lawsuits, financial penalties and manufacturing facilities getting shut down. The ultimate result could be officers getting indicted, the company suffering economic instabilities, staff downsizing, and possibly eventual bankruptcy.

2. FDA regulations mandate the need to perform Computer System Validation and these regulations have the impact of law. Failing an FDA audit can result in FDA inspectional observations ("483s") and warning letters. Failure to take corrective action in a timely manner can result in shutting down manufacturing facilities, consent decrees, and stiff financial penalties. Again, the ultimate result could be loss of jobs, indictment of responsible parties (usually the officers of a company), and companies suffering economic instabilities resulting in downsizing and possibly eventual bankruptcy.

4.4 Relationship of Computer System Validation to the Software Development Life Cycle

Computer System Validation is carried out through activities that occur throughout the entire Software Development Life Cycle (SDLC). The "V Diagram" is widely used in the IT literature to emphasize the importance of testing and testing related activities at every step in the SDLC. The V-diagram is really a recasting of the oft-criticized "Waterfall" model of the SDLC. In fact the phases in the Waterfall Model are essentially the life cycle phases that appear on the left-hand side of the V-diagram. The V-diagram emphasizes the need for various forms of testing to be part of every step along the way. This avoids a "big-bang" testing effort at the very end of the process, which has been one of the main criticisms associated with the Waterfall model (or the way some have people have interpreted the Waterfall model). The activities represented in the V-Diagram include Static Testing as well as Dynamic Testing activities. Static Testing (sometimes called Static Analysis) refers to inspections, walkthroughs, and other review/analysis activities that can be performed without actually executing the software. In Dynamic Testing, the software is actually executed and compared against expected results. While many IT people use the term "testing" to mean dynamic testing, both dynamic and static testing activities are used in Computer System Validation to help ensure that the results of successive steps in the SDLC correctly fulfill the intentions of the previous step.

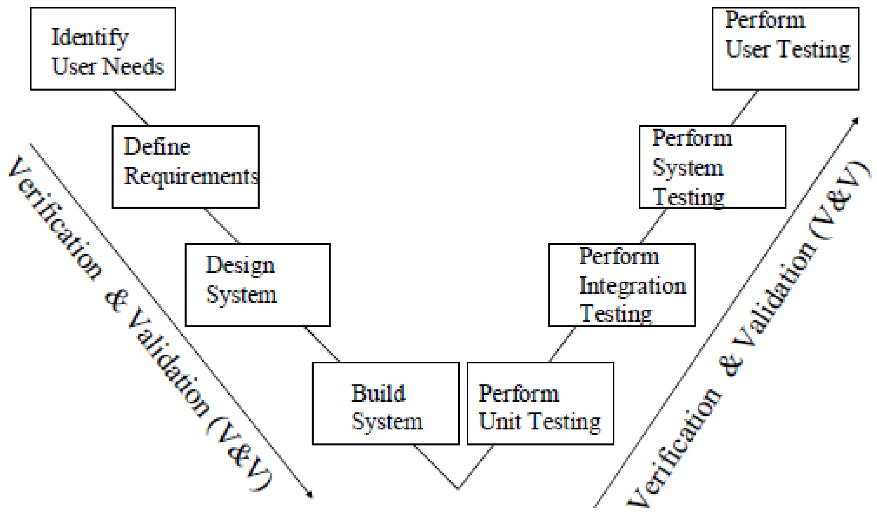


Figure 4-1: "V" Model

In companies regulated by the FDA and other regulatory bodies throughout the world, the term Validation is often used interchangeably with Computer System Validation when discussing the activities required to demonstrate that a software system meets its intended purpose.

4.5 CSV has actually extended the V-Model & put a more user driven spin on it

- Computer System Validation is driven by the "User". That is the organization choosing to apply the software to satisfy a business need is accountable for the Validation of that software. While the software supplier, the IT organization, the QA organization, and consultants can play important roles in a Computer System Validation, it is the User organization that is responsible for seeing that documented evidence supporting the Validation activities is accumulated.
- The User must write "User Requirements Specifications" (URS) to serve as the basis for a Computer System Validation. The URS provides the requirements the Computer System must fulfill for meeting business needs. A Computer System Validation cannot be done unless such a URS exists.
- The Supplier of the Computer System should provide Functional Specifications and Design Specifications, which satisfy the URS. Where such Specifications are not available for an existing system, they are sometimes developed by "reverse engineering" the functionality of the system.
- Users are involved in every step of the process (deeply involved for custom development, less for package based systems).

A three level-structure is imposed on User Testing:

The Installation Qualification or IQ focuses on testing that the installation has been done correctly.
The Operational Qualification or OQ focuses on testing of functionality in the system installed at the User site.
The Performance Qualification or PQ focuses on testing that users, administrators, and IT support people trained in the SOPs can accomplish business objectives in the production environment even under worst case conditions.

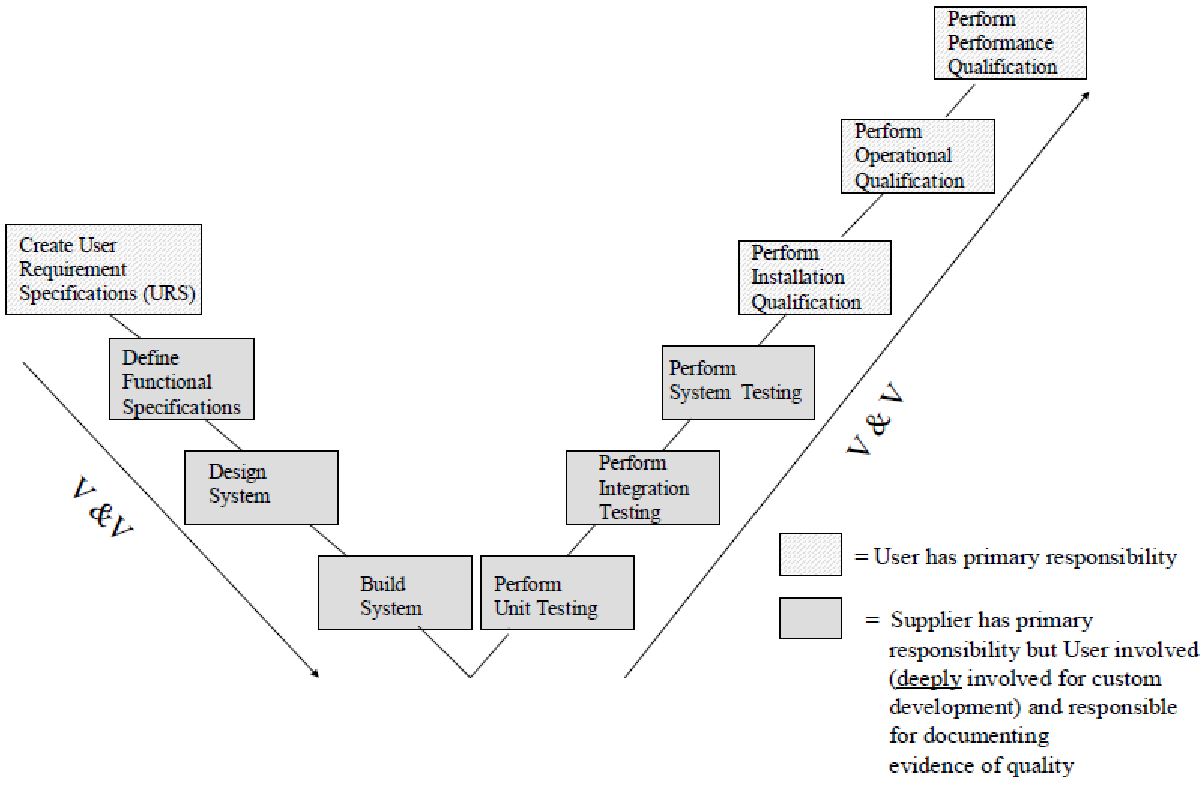

Create User Requirement Specifications (URS)
Define Functional Specifications
Design System
Build System
Perform Unit Testing
Perform Integration Testing
Perform System Testing
Perform Installation Qualification
Perform Operational Qualification
Perform Performance Qualification
V &V
V & V
= User has primary responsibility
= Supplier has primary responsibility but User involved (deeply involved for custom development) and responsible for documenting evidence of quality

Figure 4-2: CSV in FDA Regulated Industries

4.6 Relationship between Computer System Validation and 21 CFR Part 11

In 1997, the FDA added rule 21 CFR Part 11 to the Code of Federal Regulations. This regulation introduces specific controls on the use of electronic records and includes strict administrative controls on electronic signatures. These controls deal with:
1. Making electronic records suitable for supplanting paper records.
2. Making an electronic signature as secure and legally binding as a handwritten signature.
Regardless of whether or not a company uses electronic signatures, 21 CFR Part 11 impacts all companies that use computer systems that create records in electronic form associated with the GxP environment. All computer systems in this category must have technical and administrative controls to ensure:

1. The ability to generate accurate and complete copies of records
2. The availability of time-stamped audit trails
3. The protection of records to enable accurate and ready retrieval
4. Appropriate system access and authority checks are enforced

From the point of view of Computer System Validation, 21 CFR Part 11 has two key impacts.

First, it affirms that the FDA expects all computerized systems with GxP electronic records to be validated (just in case this was not obvious before). Secondly, 21 CFR Part 11 says that when we do a Validation of a particular Computer System, items 1 through 4 above automatically become part of the requirements for the System. This means that every Computer System
Validation must assess whether the system being validated satisfies requirements 1 through 4 above and must identify deviations, if any, and corrective actions. Since FDA regulated companies are anxious to avoid deviations in their Validations wherever possible, most companies in the Life Science sector are currently in a proactive mode of assessing all of their systems for 21 CFR Part 11 compliance and addressing deviations through procedural remediation, technical remediation *(Example: software upgrades)*, or replacement of non-compliant systems with 21 CFR Part 11 compliant systems.

GxP is an umbrella term that covers:

- **GMP:** Good Manufacturing Practice (sometimes called Current Good Manufacturing Practice or cGMP)
- **GLP:** Good Laboratory Practice
- **GCP:** Good Clinical Practice

The GAMP Forum (Good Automated Manufacturing Processes Forum) focuses on the application of GxP to the IT environment. The GAMP Guide for Validation of Automated Systems is said to be the most widely used, internationally accepted, guideline for validation of computer systems.

4.7 Summary on CSV

A Computer System Validation is a set of activities that FDA Regulated companies must conduct for each of their GxP sensitive computer systems. The objective of these activities is to document evidence that each computer system will fulfill its intended purpose in a GxP production, laboratory, or research operation. The intention is to avoid software problems that could have serious impact. Dynamic testing of the software is an important part of the Computer System Validation. But Computer System Validation is more than just this type of testing. Computer System Validation requires a comprehensive set of equally important static testing activities that need to be conducted throughout the SDLC. This includes a variety of analyses, audits, walkthroughs, reviews, and traceability exercises. Documentation must be accumulated that demonstrates that these activities have been performed effectively.

Today, the term Computer System Validation refers specifically to the technical discipline used in the Life Sciences sector to help ensure that software systems meet their intended requirements. Through its regulations/guidance on Computer System Validation, the FDA has shaped IT testing and analysis processes to match the needs and requirements of the industries it governs. As a result, Computer System Validation has become an integral part of doing business in FDA regulated environments. It should be noted, however, that significant progress has been made in achieving consistency and harmonization between FDA regulations/guidance on Computer System Validation and relevant international IT standards and best practices. It is likely that the future will see convergence of Computer System Validation terminology and techniques as a common technical discipline across other industry sectors as well.

4.8 Validation Script for the Module Loading

Installation Qualification:

ID	Parameters	Acceptable
I1	Operating System: Windows Server 64 bit	YES
I2	Software: Java JDK 1.6.0, Documentum Foundation Classes 6.5, DQL	YES
I3	Enterprise Content Management system: Documentum 6.5 Content Server	YES
I4	Integrated Development Environment: Eclipse Galileo	YES
I5	Connection to Server: Remote Console (mstsc)	YES
I6	File System: New Technology File System (local)	YES

Successful completion of the preceding activities and checks indicates that this environment and Softwares has been satisfactorily installed. This has *PASSED the Installation Qualification* procedure and may now be *submitted for Operational Qualification*.

Cases:	I1 – I6		
IQ Completed By:	R. Sathish Kumar	Date:	18-MAR-2015
IQ Approved By:	[illegible]	Date:	19-MAR-2015

Operational Qualification:

If Document MATCH then CHECK-IN
ELSE (NOT MATCH)
IMPORT

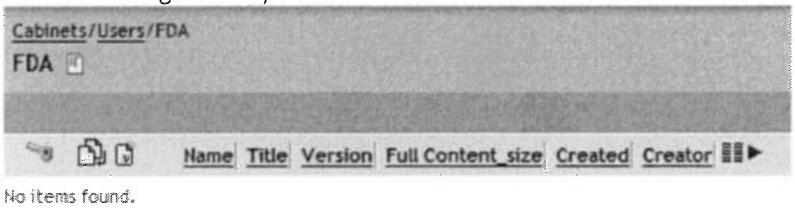
Note: The above conditions includes document extension.

Exceptional Case: If the document present in File System but the document information not present in comma separated text file, then this file will not be witnessed anywhere.

For Example:

1. *In Scenario-1 Doc-4.pdf present only in File System. This will not be witnessed anywhere in log.*
2. *In Scenario-2 Doc-7.pdf present in File System. Doc-7.zip as information in comma separated text file. Due to file extension miss match, this will result in Warning.*

Content Management System: Initial State


Cabinets/Users/FDA
FDA

Name	Title	Version	Full Content_size	Created	Creator

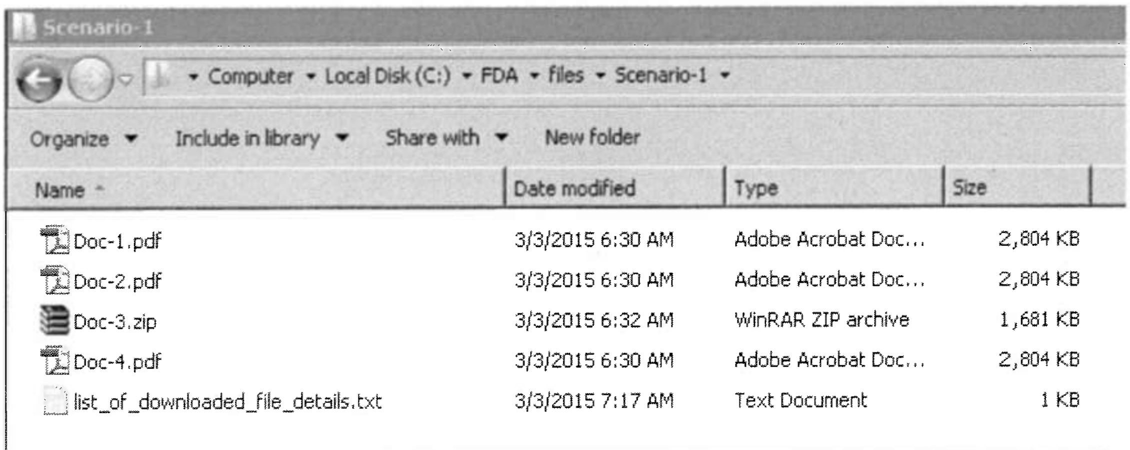
No items found.

Successful completion of the preceding activities and checks indicates that this tool has been satisfactorily developed. This has *PASSED the Operational Qualification procedure* and may now be *submitted for Performance Qualification*.

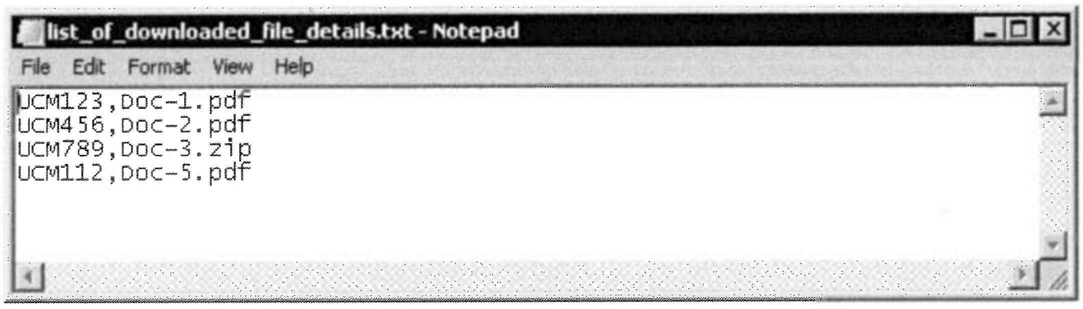
Cases:	O1 – O2		
OQ Completed By:	R. Sathish Kumar	Date:	19-MAR-2015
OQ Approved By:	[illegible]	Date:	20-MAR-2015

ID	01	Case	Scenario-1

Input: File System



Name	Date modified	Type	Size
Doc-1.pdf	3/3/2015 6:30 AM	Adobe Acrobat Doc...	2,804 KB
Doc-2.pdf	3/3/2015 6:30 AM	Adobe Acrobat Doc...	2,804 KB
Doc-3.zip	3/3/2015 6:32 AM	WinRAR ZIP archive	1,681 KB
Doc-4.pdf	3/3/2015 6:30 AM	Adobe Acrobat Doc...	2,804 KB
list_of_downloaded_file_details.txt	3/3/2015 7:17 AM	Text Document	1 KB

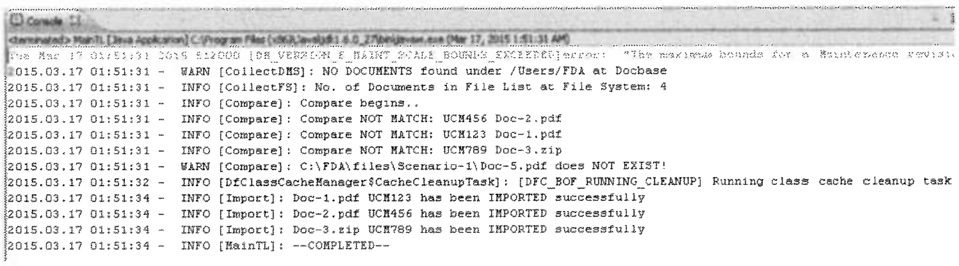

list_of_downloaded_file_details.txt - Notepad
File Edit Format View Help

```
UCM123,Doc-1.pdf
UCM456,Doc-2.pdf
UCM789,Doc-3.zip
UCM112,Doc-5.pdf
```

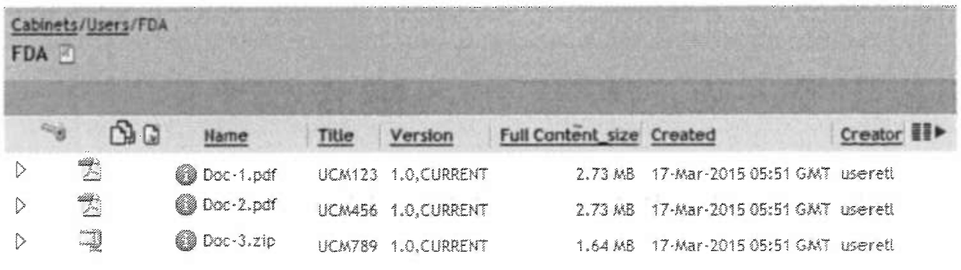
Expected	UCM123,Doc-1.pdf - Imported UCM456,Doc-2.pdf - Imported UCM789,Doc-3.zip - Imported UCM112,Doc-5.pdf - Warning

Actual

Output: Content Management System



```
<terminated> MainTL [Java Application] C:\Program Files (x86)\Java\jdk1.6.0_27\bin\javaw.exe (Mar 17, 2015 1:51:31 AM)
Tue Mar 17 01:51:31 2015 512000 [DM_VERSION_E_MAINT_SCALE_BOUNDS_EXCEEDED]error:  "The maximum bounds for a Maintenance revisi
2015.03.17 01:51:31 -  WARN [CollectDMS]: NO DOCUMENTS found under /Users/FDA at Docbase
2015.03.17 01:51:31 -  INFO [CollectFS]: No. of Documents in File List at File System: 4
2015.03.17 01:51:31 -  INFO [Compare]: Compare begins..
2015.03.17 01:51:31 -  INFO [Compare]: Compare NOT MATCH: UCM456 Doc-2.pdf
2015.03.17 01:51:31 -  INFO [Compare]: Compare NOT MATCH: UCM123 Doc-1.pdf
2015.03.17 01:51:31 -  INFO [Compare]: Compare NOT MATCH: UCM789 Doc-3.zip
2015.03.17 01:51:31 -  WARN [Compare]: C:\FDA\files\Scenario-1\Doc-5.pdf does NOT EXIST!
2015.03.17 01:51:32 -  INFO [DfClassCacheManager$CacheCleanupTask]: [DFC_BOF_RUNNING_CLEANUP] Running class cache cleanup task
2015.03.17 01:51:34 -  INFO [Import]: Doc-1.pdf UCM123 has been IMPORTED successfully
2015.03.17 01:51:34 -  INFO [Import]: Doc-2.pdf UCM456 has been IMPORTED successfully
2015.03.17 01:51:34 -  INFO [Import]: Doc-3.zip UCM789 has been IMPORTED successfully
2015.03.17 01:51:34 -  INFO [MainTL]: --COMPLETED--
```

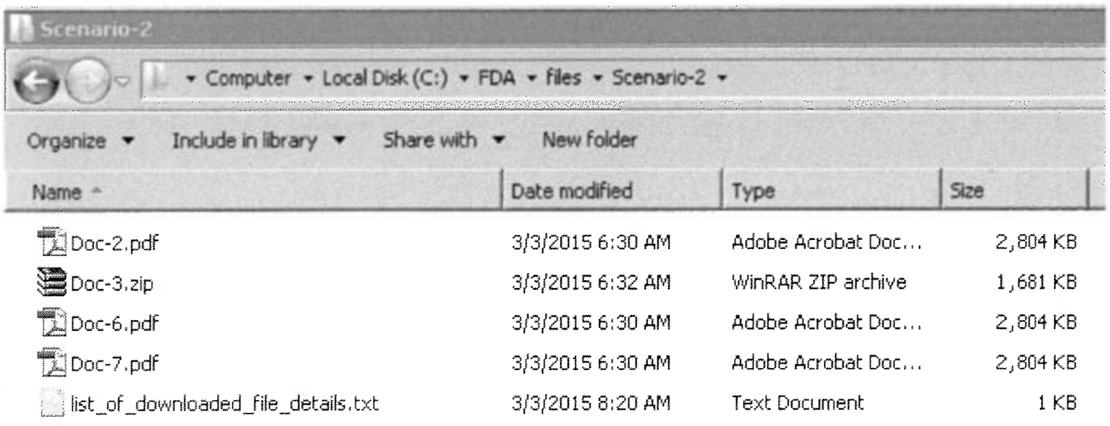


Name	Title	Version	Full Content_size	Created	Creator
Doc-1.pdf	UCM123	1.0,CURRENT	2.73 MB	17-Mar-2015 05:51 GMT	useretl
Doc-2.pdf	UCM456	1.0,CURRENT	2.73 MB	17-Mar-2015 05:51 GMT	useretl
Doc-3.zip	UCM789	1.0,CURRENT	1.64 MB	17-Mar-2015 05:51 GMT	useretl

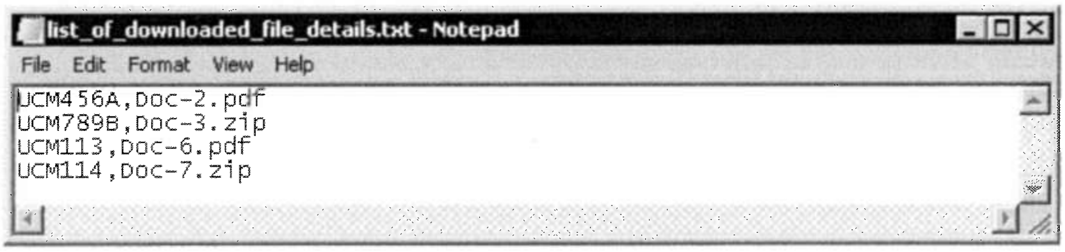
Result	PASS	Sign	R. Sathish Kumar

ID	02	Case	Scenario-2

Input: File System

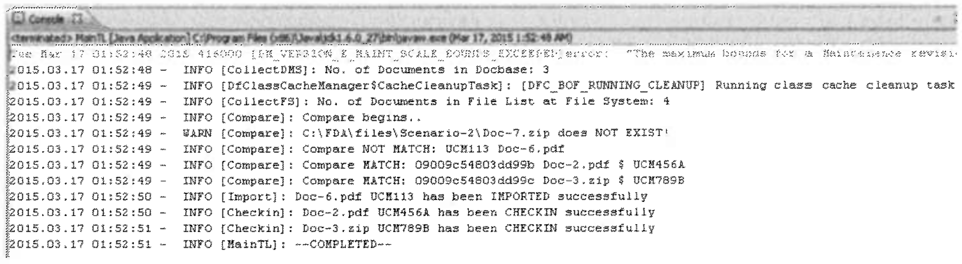


Name	Date modified	Type	Size
Doc-2.pdf	3/3/2015 6:30 AM	Adobe Acrobat Doc...	2,804 KB
Doc-3.zip	3/3/2015 6:32 AM	WinRAR ZIP archive	1,681 KB
Doc-6.pdf	3/3/2015 6:30 AM	Adobe Acrobat Doc...	2,804 KB
Doc-7.pdf	3/3/2015 6:30 AM	Adobe Acrobat Doc...	2,804 KB
list_of_downloaded_file_details.txt	3/3/2015 8:20 AM	Text Document	1 KB

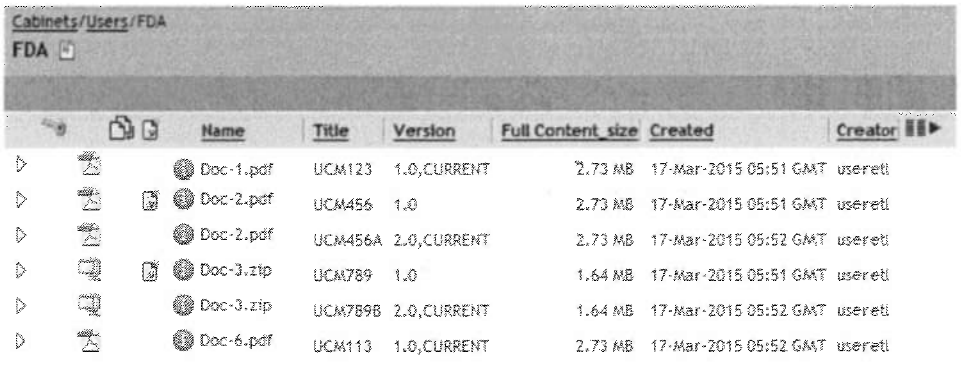

list_of_downloaded_file_details.txt - Notepad
File Edit Format View Help

```
UCM456A,Doc-2.pdf
UCM789B,Doc-3.zip
UCM113,Doc-6.pdf
UCM114,Doc-7.zip
```

Expected	UCM456A,Doc-2.pdf - Check-in UCM789B,Doc-3.zip - Check-in UCM113,Doc-6.pdf - Import UCM114,Doc-7.zip - Warning
Actual	Output: Content Management System


Console
<terminated> MainTL [Java Application] C:\Program Files (x86)\Java\jdk1.6.0_27\bin\javaw.exe (Mar 17, 2015 1:52:48 AM)

```
Tue Mar 17 01:52:48 2015 416000 [DM_VERSION_E_MAINT_SCALE_BOUNDS_EXCEEDED]error:  "The maximum bounds for a Maintenance revisi
2015.03.17 01:52:48 -  INFO [CollectDMS]: No. of Documents in Docbase: 3
2015.03.17 01:52:49 -  INFO [DfClassCacheManager$CacheCleanupTask]: [DFC_BOF_RUNNING_CLEANUP] Running class cache cleanup task
2015.03.17 01:52:49 -  INFO [CollectFS]: No. of Documents in File List at File System: 4
2015.03.17 01:52:49 -  INFO [Compare]: Compare begins..
2015.03.17 01:52:49 -  WARN [Compare]: C:\FDA\files\Scenario-2\Doc-7.zip does NOT EXIST!
2015.03.17 01:52:49 -  INFO [Compare]: Compare NOT MATCH: UCM113 Doc-6.pdf
2015.03.17 01:52:49 -  INFO [Compare]: Compare MATCH: 09009c54803dd99b Doc-2.pdf $ UCM456A
2015.03.17 01:52:49 -  INFO [Compare]: Compare MATCH: 09009c54803dd99c Doc-3.zip $ UCM789B
2015.03.17 01:52:50 -  INFO [Import]: Doc-6.pdf UCM113 has been IMPORTED successfully
2015.03.17 01:52:50 -  INFO [Checkin]: Doc-2.pdf UCM456A has been CHECKIN successfully
2015.03.17 01:52:51 -  INFO [Checkin]: Doc-3.zip UCM789B has been CHECKIN successfully
2015.03.17 01:52:51 -  INFO [MainTL]: --COMPLETED--
```



Name	Title	Version	Full Content_size	Created	Creator
Doc-1.pdf	UCM123	1.0,CURRENT	2.73 MB	17-Mar-2015 05:51 GMT	useretl
Doc-2.pdf	UCM456	1.0	2.73 MB	17-Mar-2015 05:51 GMT	useretl
Doc-2.pdf	UCM456A	2.0,CURRENT	2.73 MB	17-Mar-2015 05:52 GMT	useretl
Doc-3.zip	UCM789	1.0	1.64 MB	17-Mar-2015 05:51 GMT	useretl
Doc-3.zip	UCM789B	2.0,CURRENT	1.64 MB	17-Mar-2015 05:52 GMT	useretl
Doc-6.pdf	UCM113	1.0,CURRENT	2.73 MB	17-Mar-2015 05:52 GMT	useretl

Result	PASS	Sign	R. Sathish Kumar

Performance Qualification:

The cases defined in OQ includes Performance Qualification cases. *For Example: Different set of file extensions .pdf and .zip. ZIP format – bulk size files.* If OQ passes, then PQ can be marked as pass.

Performance: For Import, O (no. of files). For Check-in, O (size of files).

Successful completion of the preceding activities and checks indicates that this tool has been satisfactorily developed. This has *PASSED the Performance Qualification procedure* and may now be *released to use.*

Cases:	O1 – O2		
PQ Completed By:	R. Sathish Kumar	Date:	20-MAR-2015
PQ Approved By:	[illegible]	Date:	20-MAR-2015

Chapter 5: Log Manager

5.1 Introduction

Logging is an important part of development life cycle. It's an important debugging and auditing tool. Information in context to the application can be logged into a file that can be analyzed later or can provide an important medium of troubleshooting errors encountered during application development.

5.2 Logger for ET (log4j)

Log4j is a Java based logging utility primarily used as a debugging tool. It's an open source project of Apache Software Foundation that provides a reliable, fast and extensible logging library for Java. Log4j is highly configurable through external configuration files at runtime.

Log4j mainly consists of 3 parts
1. **Loggers:** Responsible for capturing logging information.
2. **Appenders:** Responsible for publishing logging information to various preferred destinations.
3. **Layouts:** Responsible for formatting logging information in different styles.

We can set different levels for Loggers via log4j configuration. The standard levels are 'DEBUG', 'INFO, 'WARN, 'ERROR' and 'FATAL' and the levels are hierarchical. So to say if we set the level to 'DEBUG' all the messages with level 'INFO, 'WARN, 'ERROR' and 'FATAL' will also show up. By default the root logger level is set to 'DEBUG'.

One of the great feature of setting up logging via Log4j is that is allows the messages to be printed to various sources that we specify like console, files, remote sockets, JMS etc. This output destination is called appender and we can attach a logger to multiple appenders. So we could have a same logging statement to be printed to a console as well as to a log file.

Also we can change the output format of the logging by associating a layout with the appender(s). The output format or the layout of a logged message is determined by 'PatternLayout' that can be configured in log4j file.

For Example: In ConsoleAppender, all the logger statements will be printed on console. RollingFileAppender direct the message to a log file.
Log file name can be set with the extension '.log' and size of the log file can be set in 'KBs'.

5.3 Logger for L (DfLogger)

Dflogger is a Documentum DFC class that can be used to enable logging from DFC Applications hence allowing the Documentum developers to troubleshoot issues and monitor logs in much better way.

To log a message in the log file, we need to import the package within which DfLogger class resides and then insert a call to the DfLogger in the class which falls under the same package for which logging has been enabled in log4j.properties file. The signature of the method is as follows:

DfLogger.debug(Object arg0, String arg1, String[] arg2, Throwable arg3)

Object arg0 : specifies the class for which the debug message is being logged. Usually, the DfLogger class is called from the class for which we log the message and hence we use the 'this' keyword. It refers to the current class instance.
String arg1: the message to log

String[] arg2: parameters to use when formatting message
Throwable arg3: a throwable to log, can be used to print a stack trace to the log file in case of an exception.

Ex: DfLogger.debug(this,"The Debug Message",null,dfe);

This way we can log all the necessary messages via DfLogger.

A good practice is to just create a single method in a utility class that can be call from any other class and used for logging messages passed to that method as a parameter. This has been implemented in FDA Web Content Mining Project.

For Documentum projects, DfLogger is an efficient and recommended way of logging and tracing messages throughout the life of an application whether those messages are logged for informational purpose, auditing purpose or even critical errors tracking. We can define logging and its type for a part of a package or entire package and we can also define multiple appenders and have different output format for each appender.

Summary

Thus using Selenium Locating Techniques and Documentum DFC; an efficient, flexible and extensible technical solution for FDA Web Content ETL has been built and validated. This solution is in compliance with FDA ICH standards.

In Simpler,
Documentum Web Publisher to FDA Site!
FDA Site to Documentum Content Management system!
But both the Docbases are different - FDA and us!

The name Selenium comes from Jason Huggins in an email, mocking a competitor named Mercury, saying that you can cure mercury poisoning by taking selenium supplements.
Selenium *Supplements* cured mercury poisoning!
Selenium *Web driver* made easier the FDA Web Content ETL!

FDA Web Content Mining is a useful project that can help any industrial expert, who are part of pharma / biotech/medical device industry and need to keep themselves updated with latest inputs from FDA, with just zero manual effort.

Directions for Future Work

- Adding to Document Name & Number, Document downloaded URL also can be loaded in Description attribute of document in Content Management System. This will helps incase if document number and document name are same.
- Detailed email notification can be send to list of registered users and admin at the end of execution and/or at any interruptions.
- Input text format files can be replaced with XML files.
- Overriding Hash Map class – Oracle built, can help to track about to add/added/overridden duplicate keys during run time.
- Extracted content can be stored in shared drive and 'Loading' module can be deployed in method server & triggered from CMS in-built jobs.
- Rather trigger ETL from in-line command batch file, a Service can be created and scheduled in Windows jobs.
- Adding to current Content Management System – Documentum, providing options to other popular CMSs like Share point, File Net may lead a way for "Project to Product".

Bibliography

Books:

1. Pawan Kumar. Documentum 6.5 Content Management Foundations. Birmingham – Mumbai: Packt Publishing, 2007.
2. Ponnaiah, Paulraj. Data Warehousing Fundamentals. Wiley-Student Edition, 2001.
3. Tan, Pang-Ning. Introduction to Data Mining. Pearson Education, 2006.
4. Elmasri, and Navathe. Fundamentals of Database Systems. Pearson Education, 2007.
5. Bass, Len. Software Architecture in Practice. Pearson Education, 3rd Ed.
6. Larman, Craig. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Pearson Education, 2004.
7. Sahni, Sartaj. Data Structures, Algorithms and Application in C++. MGHISE, 2000.
8. Pressman, R.S. Software Engineering: A Practitioner's Approach. MGHISE, 2010.
9. Paul C Jorgenson. Software Testing: A Craftsman's Approach. CRC Press, 3rd Ed.
10. Raman, Meenakshi, and Sangeeta Sharma. Technical Communication: Principles and Practice. Oxford University Press, 2011.

Scholarly journal articles:

11. Bing Liu. Web Content Mining. http://www.cs.uic.edu/~liub/WebContentMining.html, 2005.
12. S. Jeyalatha, B. Vijayakumar, Munawwar Firoz. Design and Implementation of a Tool for Web Data Extraction and Storage using Java and Uniform Interface. International Journal of Computer Applications (2011): Volume 22 (0975 – 8887).
13. James Clark and Steve DeRose. W3C XPath Specifications. http://www.w3.org/TR/xpath/.
14. N. Freed, N. Borenstein. MIME Format specification, http://www.ietf.org/rfc/rfc2045.txt.

Conference proceedings:

15. Patil, N., Shreya Patankar, Chhaya Das. A Survey on Web Content Mining and Extraction of Structured and Semistructured Data. First International Conference on Emerging Trends in Engineering and Technology, 2008. July 2008. ICETET, Nagpur, India. IEEE, 10.1109/ICETET.2008.251.
16. S.Jeyalatha, B. Vijayakumar, Zainab A. S. Design Considerations for a Data Warehouse in an Academic Environment. Oct 2010. World Academy of Science, Engineering and Technology.

Technology-Product documentations:

17. Selenium. WebDrivers and Locating Techniques. http://www.seleniumhq.org/docs/.
18. EMC^2. Documentum Foundation Classes. http://community.emc.com/.
19. W3C. XPath Tutorial. http://www.w3schools.com/xpath/default.asp.
20. List of MIME Types. http://reference.sitepoint.com/html/mime-types-full.

Policies and Standards:

21. FDA. FDA Website Policies. 2015. http://www.fda.gov/AboutFDA/AboutThisWebsite/WebsitePolicies/default.htm.
22. FDA. General Principles of Software Validation. 2014. http://www.fda.gov/RegulatoryInformation/Guidances/ucm126954.htm.
23. STSV Consulting. Computer System Validation - It's More Than Just Testing. http://www.stsv.com/pdfs/STS_CSV_article.pdf.
24. W3C. World Wide Web Consortium. http://www.w3.org/.
25. Programming Standards. http://en.wikipedia.org/wiki/Naming_convention_(programming).

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
WORK-INTEGRATED LEARNING PROGRAMMES DIVISION

Checklist of items for the Final Dissertation Report
This checklist is to be attached as the last page of the report.

This checklist is to be duly completed, verified and signed by the student.

1.	**Is the final report neatly formatted with all the elements required for a technical Report?**	[✓ Yes] / No	
2.	Is the Cover page in proper format as given in Annexure A?	[✓ Yes] / No	
3.	Is the Title page (Inner cover page) in proper format?	[✓ Yes] / No	
4.	(a) Is the Certificate from the Supervisor in proper format? (b) Has it been signed by the Supervisor?	[✓ Yes] / No [✓ Yes] / No	
5.	Is the Abstract included in the report properly written within one page? Have the technical keywords been specified properly?	[✓ Yes] / No [✓ Yes] / No	
6.	Is the title of your report appropriate? **The title should be adequately descriptive, precise and must reflect scope of the actual work done.** Uncommon abbreviations / Acronyms should not be used in the title	[✓ Yes] / No	
7.	Have you included the List of abbreviations / Acronyms?	[✓ Yes] / No	
8.	Does the Report contain a summary of the literature survey?	[✓ Yes] / No	
9.	Does the Table of Contents include page numbers? (i). Are the Pages numbered properly? (Ch. 1 should start on Page # 1) (ii). Are the Figures numbered properly? (Figure Numbers and Figure Titles should be at the bottom of the figures) (iii). Are the Tables numbered properly? (Table Numbers and Table Titles should be at the top of the tables) (iv). Are the Captions for the Figures and Tables proper? (v). Are the Appendices numbered properly? Are their titles appropriate ?	[✓ Yes] / No [✓ Yes] / No [✓ Yes] / No [✓ Yes] / No [✓ Yes] / No Yes / [✓ No]	 NA
10.	Is the conclusion of the Report based on discussion of the work?	[✓ Yes] / No	
11.	Are References or Bibliography given at the end of the Report? Have the References been cited properly inside the text of the Report? Are all the references cited in the body of the report	[✓ Yes] / No Yes / [✓ No] Yes / [✓ No]	 NA NA
12.	Is the report format and content according to the guidelines? The report should not be a mere printout of a Power Point Presentation, or a user manual. Source code of software need not be included in the report.	[✓ Yes] / ~~No~~	

Declaration by Student:
I certify that I have properly verified all the items in this checklist and ensure that the report is in proper format as specified in the course handout.

[illegible]
Signature of the Student

Place: Chennai

Date: 26-Mar-2015

Name: Sri Kumaran. T

ID No.: 2013HT12504