Team Learning Session Meeting Notes: **LOGS AND LOGGING IN CORE**

TOPICS DISCUSSED:

S.NO	Topic
1	Configuration
2	Environment Specific logging
3	Other important info

Date	**06-08-2015**
Topics Covered by:	Sudipta Sarkar
Document Compiled by:	Krishna Bodduluri
Document Version	1.0

- How Logs and logging was configured in our application?

  LogConfigUtil is the Key program to load the log4j.xml configuration file. These 2 are they important files interms of Logs configuration.

LogConfigUtil.java

```
1  package com.wellsfargo.core.common.dp.util;
2
3  import java.net.URL;
13
14 public class LogConfigUtil {
15     private static Logger log = Logger.getLogger(LogConfigUtil.class);
16     private static final String LOG_CFG_FILE = "log4j2.xml";
17     private static final String LOG_PROPERTIES_FILE = "loggingtimer.properties";
18     private static final String PROPERTY_KEY_WATCHINTERVAL = "watchInterval";
19     private static boolean isInitialized = false;
20
21     public static void configAndWatch(){
22         if(!isInitialized){
23             isInitialized = true;
24             /*URL url = LogConfigUtil.class.getClassLoader().getResource(LOG_CFG_FILE);
25             long delayInterval = getDelayTime();
26             if(delayInterval > 0){
27                 org.apache.log4j.xml.DOMConfigurator.configureAndWatch(url.getPath(), delayInterval);
28                 log.info("Log4j.xml location: " + url.getPath() + "  -  Watching interval: " + delayInterval);
29                 log.info("DOMConfigurator loaded from: "+ org.apache.log4j.xml.DOMConfigurator.class.getProtectionDomain().getCodeSource().getLocation());
30
31             }*/
32             ((LoggerContext) LogManager.getContext(false)).getConfiguration().addListener(new CustomLoggerConfigListener());
```

- Is there a separate logging process for UI and service/persistence layer?

  There is NO separate logging process for UI or Persistence layers. Logs configuration is all under log4j.xml

- What's the version of Log4j framework we are using?

  Log4j- 1.2.14. Refer to pom.xml file

- What's the purpose of log4j2.xml?

  For 4.15, we are using log4j2.xml. Log4j2.xml and lg4j.xml are very similar configuration files, that has all the info about appenders

log4j2.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<Configuration verbose="false" status="ERROR" monitorInterval="6" >
    <Appenders>

    <RollingFile name="HibernateAppender" fileName="${sys:logFilePath}/hibernate.log" filePattern="${sys:logFilePath}/hibernate.log.%i">
        <Policies>
                <SizeBasedTriggeringPolicy size="10 MB"/>
            </Policies>
        <DefaultRolloverStrategy max="10"/>
        <PatternLayout>
                <Pattern>%d{ISO8601} [%t] %p %c %x - %m\r\n </Pattern>
        </PatternLayout>
    </RollingFile>

    <RollingFile name="WFDataValidationAppender" fileName="${sys:logFilePath}/wf-datavalidation.log" filePattern="${sys:logFilePath}/wf-datavalidation.log.%i">
        <Policies>
                <SizeBasedTriggeringPolicy size="10 MB"/>
            </Policies>
        <DefaultRolloverStrategy max="10"/>
        <PatternLayout>
```

- What's the important log file of all the log files?
  Out of all the log files, wellsfargo.log file is the key log file

- What's the file size limit for each log file? How many days will the file be saved on drive?
  For each files, we are limiting the size to 10 MB. We are saving each file for 10 days. We have these details in log4j2.xml file.

- Does the splunk copies/polls the log folder and displays the content?
  Yes. Splunk copies the log content from respective servers and displays the content.
- Is there a separate sever that we have to look for quartz jobs/listeners logs? or everything is dumped to Splunk at the end?
  Quartz jobs configured to run on 3 different servers. Logs for these servers also get logged into Splunk.
- How do we know if Splunk is available for a server?
  For most of the servers, Splunk is already configured. If the Splunk Is not configured, that says- it's not high priority to know about the logs.
- How the LogConfigUtil.Java does get initialized?
  CoreActivatorImpl.java is the key program to initialize LogConfigUtil.Java

CoreActivatorImpl.java

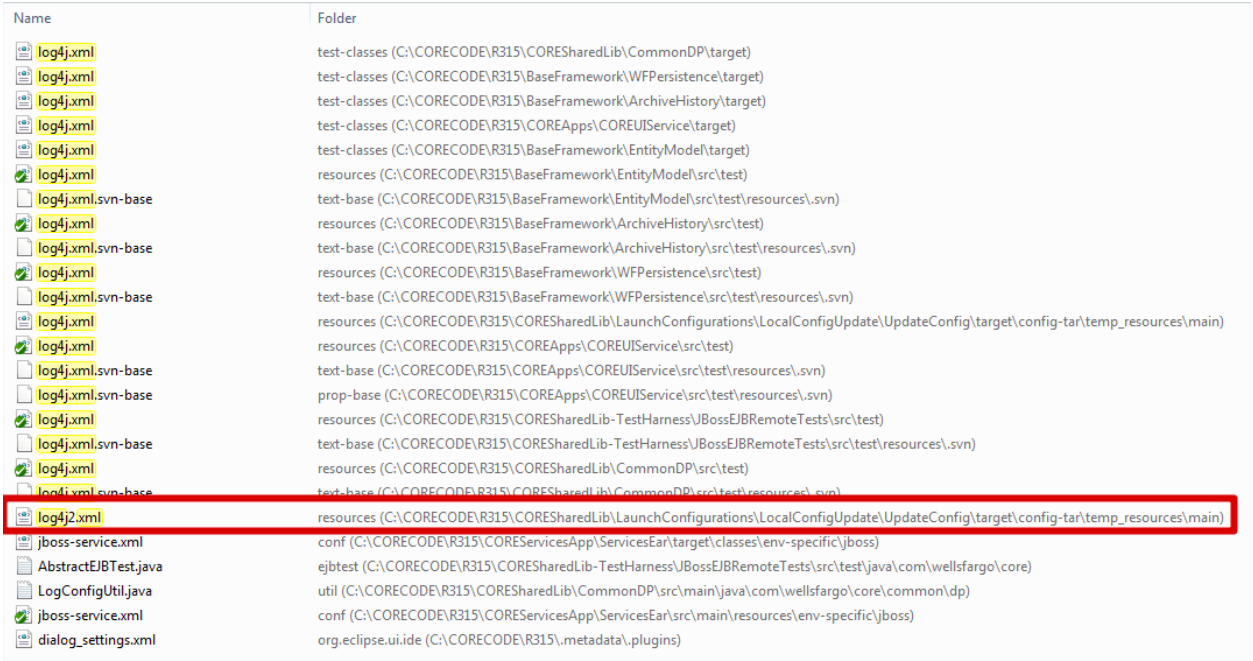
```java
/**
package com.wellsfargo.core.business.service.dp.activator;

import java.util.ArrayList;
//import com.wellsfargo.core.jca.service.WFWorkManager;


public class CoreActivatorImpl implements CoreActivator {

    @Autowired
    @Qualifier("com.wellsfargo.core.entity.service.ref.ReferenceDataEntityService")
    private ReferenceDataEntityService referenceDataEntityService;

    @Autowired
    @Qualifier("com.wellsfargo.core.common.dp.quartz.scheduler.QuartzJobScheduleService")
    protected QuartzJobScheduleService quartzJobScheduleService;

    @Autowired
    @Qualifier("com.wellsfargo.core.argentrule.integration.pipelinemigration.PipelineMigrationService")
    protected PipelineMigrationService pipelineMigrationService;

    @Autowired
    protected RegisterEventListnerService registerEventListnerService;


    private static Logger LOG = Logger.getLogger(CoreActivatorImpl.class);


    private long startTime = 0;
    private String coreAppName = null;
    private String localquartzEnableStatus = "true";
    private static boolean isStarted = false;


    public synchronized  void start(){
        if(isStarted){
            LOG.info("CoreActivator Already Initialized");
            return;
        }
        isStarted = true;
        initialize();
    }

    public void initialize() {
        SharedEnvironmentConfig.getInstance();
        PerfLogTimer perfLog = new PerfLogTimer();
        try {
            coreAppName = System.getProperty(CommonConstants.JVM_PROPERTY_APP_NAME);
            if(StringUtil.isEmptyOrNull(coreAppName)){
                coreAppName=CommonConstants.LOCAL;
            }
```

CoreActivator.java gets initialized from module-spring-context.xml file

module-spring-context.xml

```xml
<bean id="com.wellsfargo.core.business.service.dp.pricing.ajs.ProgramDetailsBusinessService" class="com.wellsfargo.core.business.service.dp.pricing.ajs.ProgramDetailsBusinessServiceImpl" />

<bean id="com.wellsfargo.core.business.service.dp.activator.CoreActivator" class="com.wellsfargo.core.business.service.dp.activator.CoreActivatorImpl" />

<bean id="com.wellsfargo.core.business.service.dp.activator.loader.ReferenceDataCacheLoader" class="com.wellsfargo.core.business.service.dp.activator.loader.ReferenceDataCacheLoaderImpl" />

<bean id="com.wellsfargo.core.business.service.dp.template.ajs.delegator.SaveInTemplateDelegator" class="com.wellsfargo.core.business.service.dp.template.ajs.delegator.impl.SaveInTemplateDelegatorImpl" />

<bean id="com.wellsfargo.core.business.service.dp.template.ajs.SaveInLoanTemplateHelper" class="com.wellsfargo.core.business.service.dp.template.ajs.SaveInLoanTemplateHelperImpl" />

<bean id="com.wellsfargo.core.business.service.dp.template.ajs.SaveDocumentTemplateBusinessService" class="com.wellsfargo.core.business.service.dp.template.ajs.SaveDocumentTemplateBusinessServiceImpl" />

<bean id="com.wellsfargo.core.business.service.dp.template.ajs.SaveInTemplateBusinessService" class="com.wellsfargo.core.business.service.dp.template.ajs.SaveInTemplateBusinessServiceImpl" />

<bean id="com.wellsfargo.core.business.service.dp.template.ajs.TemplateGroupDefinitionsBusinessService" class="com.wellsfargo.core.business.service.dp.template.ajs.TemplateGroupDefinitionsBusinessServiceImpl" />

<bean id="com.wellsfargo.core.business.service.dp.template.ajs.SaveInTemplatePrepaidHelper" class="com.wellsfargo.core.business.service.dp.template.ajs.SaveInTemplatePrepaidHelperImpl" />

<bean id="com.wellsfargo.core.business.service.dp.applytemplate.ajs.delegator.ApplyTemplateDelegator" class="com.wellsfargo.core.business.service.dp.applytemplate.ajs.delegator.impl.ApplyTemplateDelegatorImpl" />

<bean id="com.wellsfargo.core.business.service.dp.applytemplate.ajs.delegator.LoadLoanTemplateDelegator" class="com.wellsfargo.core.business.service.dp.applytemplate.ajs.delegator.impl.LoadLoanTemplateDelegatorImpl" />

<bean id="com.wellsfargo.core.business.service.dp.applytemplate.ajs.ApplyTemplateBusinessService" class="com.wellsfargo.core.business.service.dp.applytemplate.ajs.ApplyTemplateBusinessServiceImpl" />
```

- How many logger levels are we using in our app.? (like DEBUG, INFO, WARN, ERROR, FATAL)?
  Most of the times, ERROR is enabled in PRODUCTION. Depending on the severity of the issue, other options can be enabled.
- I see several log4j.xml files from tests folder of application and only one log4j.xml from modules application. So, is it only one main configuration file?
  Ignore all the log4j.xml files form tests folders. Below highlighted one is the main configuration file.

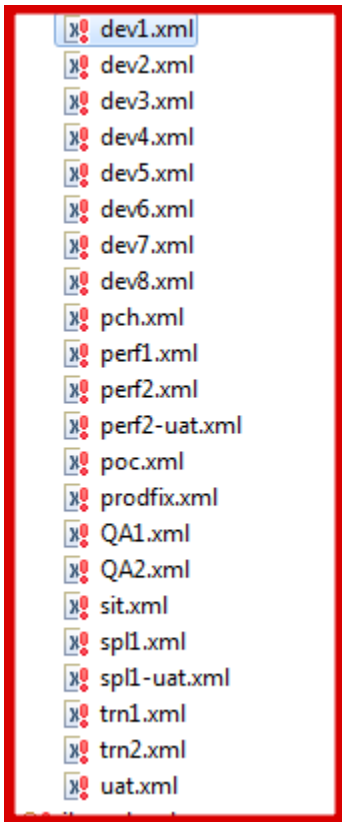


Name	Folder
log4j.xml	test-classes (C:\CORECODE\R315\CORESharedLib\CommonDP\target)
log4j.xml	test-classes (C:\CORECODE\R315\BaseFramework\WFPersistence\target)
log4j.xml	test-classes (C:\CORECODE\R315\BaseFramework\ArchiveHistory\target)
log4j.xml	test-classes (C:\CORECODE\R315\COREApps\COREUIService\target)
log4j.xml	test-classes (C:\CORECODE\R315\BaseFramework\EntityModel\target)
log4j.xml	resources (C:\CORECODE\R315\BaseFramework\EntityModel\src\test)
log4j.xml.svn-base	text-base (C:\CORECODE\R315\BaseFramework\EntityModel\src\test\resources\.svn)
log4j.xml	resources (C:\CORECODE\R315\BaseFramework\ArchiveHistory\src\test)
log4j.xml.svn-base	text-base (C:\CORECODE\R315\BaseFramework\ArchiveHistory\src\test\resources\.svn)
log4j.xml	resources (C:\CORECODE\R315\BaseFramework\WFPersistence\src\test)
log4j.xml.svn-base	text-base (C:\CORECODE\R315\BaseFramework\WFPersistence\src\test\resources\.svn)
log4j.xml	resources (C:\CORECODE\R315\CORESharedLib\LaunchConfigurations\LocalConfigUpdate\UpdateConfig\target\config-tar\temp_resources\main)
log4j.xml	resources (C:\CORECODE\R315\COREApps\COREUIService\src\test)
log4j.xml.svn-base	text-base (C:\CORECODE\R315\COREApps\COREUIService\src\test\resources\.svn)
log4j.xml.svn-base	prop-base (C:\CORECODE\R315\COREApps\COREUIService\src\test\resources\.svn)
log4j.xml	resources (C:\CORECODE\R315\CORESharedLib-TestHarness\JBossEJBRemoteTests\src\test)
log4j.xml.svn-base	text-base (C:\CORECODE\R315\CORESharedLib-TestHarness\JBossEJBRemoteTests\src\test\resources\.svn)
log4j.xml	resources (C:\CORECODE\R315\CORESharedLib\CommonDP\src\test)
log4j.xml.svn-base	text-base (C:\CORECODE\R315\CORESharedLib\CommonDP\src\test\resources\.svn)
log4j2.xml	resources (C:\CORECODE\R315\CORESharedLib\LaunchConfigurations\LocalConfigUpdate\UpdateConfig\target\config-tar\temp_resources\main)
jboss-service.xml	conf (C:\CORECODE\R315\CORESharedLib\LaunchConfigurations\LocalConfigUpdate\UpdateConfig\target\config-tar\temp_resources\main)
AbstractEJBTest.java	ejbtest (C:\CORECODE\R315\CORESharedLib-TestHarness\JBossEJBRemoteTests\src\test\java\com\wellsfargo\core)
LogConfigUtil.java	util (C:\CORECODE\R315\CORESharedLib\CommonDP\src\main\java\com\wellsfargo\core\common\dp)
jboss-service.xml	conf (C:\CORECODE\R315\COREServicesApp\ServicesEar\src\main\resources\env-specific\jboss)
dialog_settings.xml	org.eclipse.ui.ide (C:\CORECODE\R315\.metadata\.plugins)

- What about JMS logs?
  We don't have direct access to Logs that are SPECIFIC to JMS. Usually if any queue is down, Admins will take care of it. But if we want to see the logs, contact build team.
- Are the below files important? NO
  - debug.js
  - layouts.js
  - levels.js
  - log4js.js
  - log4js.json
  - logger.js

- How does the Logger picks up config information that's specific to each environment?

  We maintain configuration files that are specific to each environment. Logger framework reads info from these files and initializes the settings.



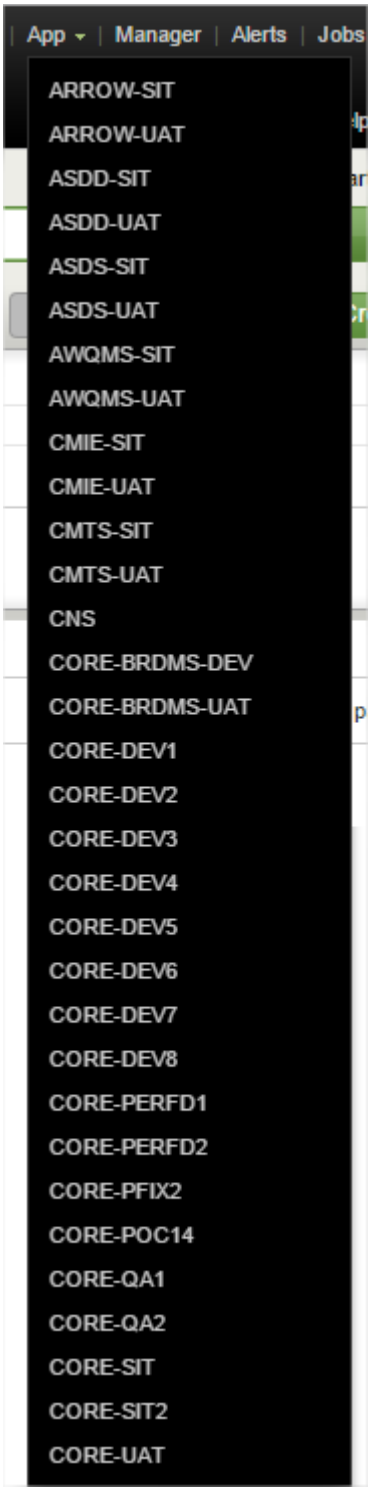
- What is log4j-jboss.xml?

  This is server related log file. This file is related to JBOSS servers log configuration.

- What is log4j-jboss.xml?

index=*coreuat* "Caused by:*Exception"
NOT com.wellsfargo.core.common.util.service.exceptions.DataValidationException
NOT com.wellsfargo.core.integration.mapper.hula.pricing.NoPricesDataValidationException
|rex "(?P<cause>Caused by: [^\r\n]+)" |rex "ERROR (?P<errorClass>[^\s]+)" |stats count by cause, errorClass, httpRequestURL |sort - count
|rex max_match=150 field="cause" "(?<split__regex>.{0,150}(?:\s|$)|.{150})" | rename split__regex as "Error_Log_Event"
|table count, Error_Log_Event , errorClass, httpRequestURL

- To Switch/view logs of to different environments, below is one of the option in Splunk.



More info from Wiki on logs:

Here is the link for Log Analysis that Paul is leading:

http://t01drw2510.dsm.tic.wellsfargo.com:8090/display/APL/CORE+Log+Analysis