

Title:	CLUWE Web Tool System Administration and Support SOP
---------------	--

Document Number:	ITLIB-SOP-000226-CLUWE Web Tool System Admin and Support	
Version: 3	Reviewers – Date (YYYY-MM-DD) and Time Approved. ITL Reviewer - Sheppard Christopher - 2017-06-14T16:44:22	
Status (at print): Effective	ITL Reviewer - Fleig David Christian - 2017-06-14T16:52:28	
Effective Date: 2017-07-14T00:02:48	Approvers – Date (YYYY-MM-DD) and Time Approved. ITL Approver - Rees John Jason - 2017-06-14T17:13:35	

If you are using a printed copy of this document, check that the version number is consistent with the current version number located in the Regulus system. This page does not display more than 25 signatures or signatures obtained outside the system. Refer to local standard operating procedures for additional information.



CONFIDENTIAL

TITLE: CLUWE WEB TOOL SYSTEM ADMINISTRATION AND SUPPORT SOP

Training Requirements

Place an "X" in the appropriate box.		
New Procedure		
Training Required	Yes <input type="checkbox"/>	No <input type="checkbox"/>
Revised Procedure		
Retraining Required	Yes <input checked="" type="checkbox"/>	No <input type="checkbox"/>

REVIEWERS

Your electronic signature indicates that you have reviewed this document and that for your area of expertise, agree it is accurate and complete.

IT Subject Matter Expert

Quality Consultant

APPROVERS

Subject Matter Expert (SME)

Your electronic signature attests that this procedure is accurate and complete.

TITLE: CLUWE WEB TOOL SYSTEM ADMINISTRATION AND SUPPORT SOP

PURPOSE

The purpose of this Standard Operating Procedure (SOP) is to describe processes and procedures necessary to maintain and support the system.

SCOPE

The following are in scope:

- CLUWE Web Tool Code Deployment
- CLUWE Web Tool Support and Maintenance

The following are out of scope:

- Processes for Infrastructure services that support the system.
- Business processes.

ACRONYMS AND DEFINITIONS

The terms and acronyms in this document are defined at their first occurrence.

DOCUMENT REVISION HISTORY

Version	Revision Date	Reason for Revision (Include CR#, if Applicable)	Revised By, Title
3	14-Jun-2017	Updated for release 3.0, CHG1074979. <ul style="list-style-type: none">• Changed sections 7.0, 7.1, 7.2.2, 7.3.2, 7.3.3, 7.4.3, 10.2, 10.4, 11.2.• Added section 12• Removed section 11.2	David Fleig, IT SME
2	21-Sep-2016	Updated for release 2.1, CHG0988306, added sections: <ul style="list-style-type: none">• 11.5.1 Version Monitoring• 11.7.1 eSignature Monitoring• 11.8.1 Lock-in-place Monitoring	John Jason Rees, IT SME
1	24 Aug 2016	Updated per CHG0783626. Changes made to the following sections: 7.1.1, 7.1.2, 7.2, 7.2.1, 7.2.3, 7.3, 7.3.1, 7.3.2 7.3.3, added New 7.2.2 updated 10.1: added row 4, ActiveMQ Brokers 11.5, 11.6, updated descriptions 11.7, 11.8, 11.9, 11.10, 11.11 added New	David Fleig, IT SME
0	18-Feb-2016	New Version	David Fleig, Technical Lead

Table of Contents

1. RISKS	5
2. CHANGE MANAGEMENT	5
2.1 CHANGE CONTROL BOARD	5
2.2 PRE-APPROVED CHANGES.....	5
2.3 VULNERABILITY PATCHES	5
3. INCIDENT MANAGEMENT.....	5
4. PROBLEM MANAGEMENT	6
5. SOURCE CODE MANAGEMENT.....	6
5.1 SOURCE CODE MANAGEMENT	6
5.2 CODE MIGRATION AND SOFTWARE INSTALLATION	6
5.3 AUTOMATED TOOL.....	6
6. REQUIRED SOFTWARE.....	6
7. DEPLOYMENT PROCESS.....	6
7.1 DEPLOYABLE ARTIFACT INFORMATION	7
7.2 DEV DEPLOYMENTS.....	8
7.2.1 Webapp Deployment.....	8
7.2.2 Service Deployment	8
7.3 QA DEPLOYMENT.....	9
7.3.1 Build the Release Code.....	9
7.3.2 Webapp Deployment.....	9
7.3.3 Service Deployment	10
7.4 PRD DEPLOYMENT	11
7.4.1 Build the Release Code.....	11
7.4.2 Webapp Deployment.....	12
7.4.3 Service Deployment	12
8. PERIODIC REVIEW	13
9. SYSTEM INVENTORY	13
10. INFRASTRUCTURE AND INTERFACE SUPPORT INFORMATION	13
10.1 INFRASTRUCTURE SUPPORT	13
10.2 INTERFACE SUPPORT	13
10.3 NECESSARY SUPPORT ACCESS	14
10.4 NECESSARY TESTING ACCESS	14
10.5 NECESSARY DEVELOPMENT ACCESS	14
11. ROUTINE ADMINISTRATION TASKS.....	14
11.1 ADD NEW ELECTRONIC SIGNATURE MEANING ENTRY	14
11.2 ADJUST THE NUMBER OF SCHEDULED SAS JOBS DISPLAYED TO THE USER	14
11.3 DISPLAY OVERALL VERSION, SIGNATURE AND FILE LOCK STATISTICS	14
11.4 SCAN THE DATABASE FOR FAILED VERSION OPERATIONS	15
11.5 SCAN THE DATABASE FOR FAILED LOCK-IN-PLACE OPERATIONS	15
11.6 SCAN THE DATABASE FOR FAILED SIGNATURE OPERATIONS	16
11.7 VERIFY THAT LOCK-IN-PLACE IS OPERATING	16

12. SERVICE MANAGEMENT TASKS	16
12.1 START THE CLUWE SERVICES.....	16
12.2 START THE CLUWE SERVICES IN DEBUG MODE	17
12.3 STOP THE CLUWE SERVICES	18
12.4 VERSION SERVICE MONITORING.....	18
12.5 SIGNATURE SERVICE MONITORING	18
12.6 VERIFY THAT CLUWE SERVICES ARE RUNNING ON A SERVER.....	19
13. REFERENCES	19

1. RISKS

The security risks involved with this system are present in the section *Security Risks* of the *CLUWE Web Tool Security Plan* document located in Regulus.

2. CHANGE MANAGEMENT

Changes to the system (hardware and software changes) and data will be made according to, ITC-SOP-Change Management-412. The tool used to manage change requests associated with this system is ServiceNow.

All required assignment groups, approval groups will be auto populated in ServiceNow once the Configuration item is selected.

2.1 Change Control Board

There is no change control board for this system.

2.2 Pre-Approved Changes

There are no pre-approved changes for this system.

2.3 Vulnerability Patches

Vulnerability patches for the jBoss web servers are managed by Web Hosting Organization (WHO). Vulnerability patches for the Isilon Network Attached Storage (NAS) cluster are managed by the NAS Platform team. The IT Support Team will work with platform teams to verify that the patches don't impact the system. The testing and results are documented in a Service Now Service Request.

3. INCIDENT MANAGEMENT

Incidents are logged via the Incident Management application in ServiceNow. Selecting the correct Configuration Item (CI) will automatically populate the incident with information indicating the business criticality and assists with the Assessment and Investigation of the incident. Every CI has a related Support Group. The CI value can also be used to route incident to the appropriate Assignment Group. Closed incidents remain available in ServiceNow as a historical resource for System Administrators to use when diagnosing/resolving future issues.

For other requests where a standard service is to be performed, a Service Request is raised. Service Requests will be logged via the Service Catalog, if a catalog item exists; if not, non-catalog Service Request items will be logged

Type of Information	Details
Priority	Priority is used for Incidents only. Service Request Service Level Agreements (SLA) will be determined on a case by case basis. Within Service Requests, there will be Standard and enhanced SLAs if appropriate. For Service Requests, the catalog item will automatically determine the SLA and no documentation are needed other than referring to the catalog item.
Assignment Group	GSS-LRLIT-GLB
Configuration Items	CLUWE

4. PROBLEM MANAGEMENT

Problem Management aims to resolve the root cause of incidents and to manage the lifecycle of a problem. In ServiceNow, the problems are created and managed by the Problem application. Problems can be created as standalone problems through analysis or linked with an incident (or group of similar incidents). Closed problem tickets remain available in ServiceNow as a historical resource for System Administrators to refer for root cause of the issues.

Problems can be flagged as deviation by CSQA.

ITC-SOP-Problem Management-389 describes how the problem tickets are utilized versus incident.

5. SOURCE CODE MANAGEMENT

5.1 Source Code Management

Source code for the application files used for development, testing and deployed to production are maintained in Redmine, which is supported by the HPC team. Apache subversion (svn) is used to commit and provide versioning control for the source code to the development (DEV), quality assurance (QA) and production environments

Microsoft Team Foundation Server (TFS) is used to store the QA and Production deployment code and artifacts. The Solution Services group performs management of the TFS software and server

5.2 Code Migration and Software Installation

Code migration is described in the Deployment Method section of this document. No software installation is required.

5.3 Automated Tool

Team Foundation Server (TFS) and SVN are used to manage source code.

6. REQUIRED SOFTWARE

The following tools are required to perform typical support activities for the system:

- Database access tool such as TOAD or Oracle SQL Developer to access the CLUWE database.
- IE 11

In addition to the support tools, the tools below are required to perform typical development activities for the system:

- An Integrated Development Environment (IDE) that supports Java 1.8 and Apache Maven.
- TFS Client Software
- SVN Software
- Internet Explorer
- SSH Software
- SFTP Software

7. DEPLOYMENT PROCESS

Deployment of the Web Front End in java WAR file format is performed using the processes defined by the Web Hosting Organization (WHO). Deployment of the CLUWE services on the QA and PRD servers is performed by the CLUWE Support Team using the processes in this document. Dev deployment of the services is performed by the CLUWE Dev Team.

7.1 Deployable Artifact Information

The table below lists the code modules deployed, their installation directory and the servers they're installed on.
<version> is the current version of the item.

Service Name	Installation Directory / Artifact Name	Shell Script	Executing Servers
Locker	/opt/cluwe/locker.adapter/locker.adapter-<version>.jar	locker.sh	DEV: cluwe-d3, cluwe-d4, cluwe-d5, cluwe-d6 QA: cluwe-q3, cluwe-q4, cluwe-q5, cluwe-q6 PRD: cluwe-p3, cluwe-p4, cluwe-p5, cluwe-p6
Signature	/opt/cluwe/signature.adapter/signature.adapter-<version>.jar	signature.sh	DEV: cluwe-d3, cluwe-d4, cluwe-d5, cluwe-d6 QA: cluwe-q3, cluwe-q4, cluwe-q5, cluwe-q6 PRD: cluwe-p3, cluwe-p4, cluwe-p5, cluwe-p6
Version	/opt/cluwe/version.adapter/version.adapter-<version>.jar	version.sh	DEV: cluwe-d3, cluwe-d4, cluwe-d5, cluwe-d6 QA: cluwe-q3, cluwe-q4, cluwe-q5, cluwe-q6 PRD: cluwe-p3, cluwe-p4, cluwe-p5, cluwe-p6
Provision	/opt/cluwe/provision/provision-<version>.jar	provision.sh	DEV: cluwe-d3, cluwe-d4, cluwe-d5, cluwe-d6 QA: cluwe-q3, cluwe-q4, cluwe-q5, cluwe-q6 PRD: cluwe-p3, cluwe-p4, cluwe-p5, cluwe-p6
ComputeJob	/opt/cluwe/computejob/computejob-<version>.jar	computejob.sh	DEV: cluwe-d5 QA: cluwe-q5 PRD: cluwe-p5
Lock In Place	/opt/cluwe/lip/cluwe-cli-<version>.jar	N/A	DEV: cluwe-d6 QA: cluwe-q6 PRD: cluwe-p6
Web Front End	cluwe.webapp-<version>.war Installed by the Web Hosting Organization	N/A	DEV: cluwe-d1, cluwe-d2 QA: cluwe-q1, cluwe-q2

			PRD: cluwe-p1, cluwe-p2
--	--	--	-------------------------

7.2 DEV Deployments

7.2.1 Webapp Deployment

Step	Details
1	Login to the Jenkins (https://swci.am.lilly.com:8443/) and open the job titled “CLUWE – Trunk”
2	Open the build to deploy from the Build History
3	Download the cluwe.webapp-<version>.war file from the “CLUWE :: UI :: Webapp” module.
4	Transfer the file to cluwe-d1.am.lilly.com and place it in the /od/cluwe/installableApp directory.
5	SSH to cluwe-d1.am.lilly.com then cd to /od/cluwe/ installableApp
6	Delete the existing cluwe.war file then rename the file transferred in step 4 to cluwe.war.
7	Execute the devappmaint.sh script to remove the existing application.
8	Execute the devappmaint.sh script to deploy the new war file.
9	Login to https://cluwe-d1.am.lilly.com to verify that the app deployed.

7.2.2 Service Deployment

The process below documents the recommended deployment steps for the development environment. Developers have the option of adopting only portions of this process for the development environment due to the fluid nature of development and troubleshooting activities.

Step	Details
1.	SSH to cluwe-q3.am.lilly.com then execute the command “ <code>sudo /etc/bcluwe_services_q</code> ” without the quotes. When prompted, enter your Lilly password.
2.	Stop any running instances of the service you’re going to deploy using the <i>Stop the CLUWE Services</i> process in this document.
3.	Change to the /home/cluwe_services/install directory and remove any existing files.
4.	Execute the following commands to download the deployment artifacts from Jenkins. Replace <version> with the version to deploy. <code>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/service/computejob/target/computejob-<version>.jar</code> <code>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/service/locker.adapter/target/locker.adapter-<version>.jar</code> <code>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/service/provision/target/provision-<version>.jar</code> <code>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/service/signature.adapter/target/signature.adapter-<version>.jar</code> <code>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-</code>

Step	Details
	%20Release/lastSuccessfulBuild/artifact/service/version.adapter/target/version.adapter-<version>.jar wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/ui/webapp/target/cluwe.webapp-<version>.war
5.	Copy the contents of the deploy/prd folder from the release branch of SVN to the /home/cluwe_services/install directory
6.	Execute the following two commands to update the service management scripts with the correct version of jar file to start. Substitute the release version for the <version> string. sed -i -- 's/ARTIFACT_VERSION="[:digit:][:digit:][:digit:]" /ARTIFACT_VERSION="<version>"/' *.sh
7.	Verify that step 6 was successful by ensuring that the ARTIFACT_VERSION value in locker.sh matches the version in the locker.adapter-<version>.jar file.
8.	Copy the jar files and shell scripts to their installation directories using the commands below: cp ~/install/locker* /opt/cluwe/locker.adapter cp ~/install/provision* /opt/cluwe/provision cp ~/install/signature* /opt/cluwe/signature.adapter cp ~/install/version* /opt/cluwe/version.adapter cp ~/install/computejob* /opt/cluwe/computejob cp ~/install/bjobs.sh /opt/cluwe/computejob cp ~/install/submitsasjob.sh /opt/cluwe/sasgsub
9.	Use the process <i>Start the CLUWE Services</i> process in this document to start the processes on their servers.

Once the service(s) file is deployed, verification should be performed using the Verification Process in the *CLUWE Disaster Recovery Plan*. If deploying a new webapp and new services, the verification only need to be performed once at the end of all the deployments.

7.3 QA Deployment

7.3.1 Build the Release Code

The results of this build are used for the Webapp and Service deployments.

1	Remove the “-SNAPSHOT” string from the version in the pom files.
2	Build the code locally to ensure it builds successfully.
	Commit the code to the release branch then start a Jenkins build for that branch.
3	Tag the code in SVN with a tag name of QA-<Maven Version>-<CHG Number> where CHG Number is the Service Now request number for the release.
4	Copy the code from the tag to a TFS directory of the same name.

7.3.2 Webapp Deployment

This process wraps steps around the WHO deployment process to archive the code and deployed war file before deployment.

Step	Details
1	SSH into cluwe-d1 and create a directory named "qa-<mmddyyyy>" where mm is the number of the current month, dd is the day of the month and yyyy is the year.
2	Login to the Jenkins (https://swci.am.lilly.com:8443/) and open the job titled "CLUWE – Release"
3	Open the build to deploy from the Build History
4	SSH into cluwe-d1 and create a directory named "qa-<mmddyyyy>" where mm is the number of the current month, dd is the day of the month and yyyy is the year.
5	Download the cluwe.webapp-<version>.war file from the "CLUWE :: UI :: Webapp" module and transfer it to the directory created in the previous step.
6	<p>Create a ServiceNow request to deploy a WebSite using the form <i>Home > Order Things > Technology Infrastructure > Web Hosting and Analytics > Website/Application Deployment</i></p> <p>Use the following values when filling out the form:</p> <p>Select appropriate platform: JBOSS</p> <p>Which environment are you deploying to? QA</p> <p>Specify Application Name: CLUWE</p> <p>Is this the first time this code has ever been deployed? No</p> <p>What server are you deploying FROM? cluwe-d1</p> <p>What server are you deploying TO? cluwe-q1</p> <p>What file(s) and/or directories require deployment (please include full path)? /od/cluwe/installableApp/<folder name from step 4>/<filename from step 5></p> <p>Where do you want the file or folders copied TO (please include full path)? /od/cluwe/installableApp/cluwe.war</p>
7	Once the service(s) file is deployed, verification should be performed using the Verification Process in the <i>CLUWE Disaster Recovery Plan</i> . If deploying a new webapp and new services, the verification only need to be performed once at the end of all the deployments.

7.3.3 Service Deployment

Step	Details
10.	SSH to cluwe-q3.am.lilly.com then execute the command "sudo /etc/bcluwe_services_q" without the quotes. When prompted, enter your Lilly password.
11.	Stop any running instances of the service you're going to deploy using the <i>Stop the CLUWE Services</i> process in this document.
12.	Change to the /home/cluwe_services/install directory and remove any existing files.
13.	<p>Execute the following commands to download the deployment artifacts from Jenkins. Replace <version> with the version to deploy.</p> <pre>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/service/computejob/target/computejob-<version>.jar</pre> <pre>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/service/locker.adapter/target/locker.adapter-<version>.jar</pre>

Step	Details
	<pre>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/service/provision/target/provision-<version>.jar</pre> <pre>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/service/signature.adapter/target/signature.adapter-<version>.jar</pre> <pre>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/service/version.adapter/target/version.adapter-<version>.jar</pre> <pre>wget --no-check-certificate -nd https://swci.am.lilly.com:8443/job/CLUWE%20-%20Release/lastSuccessfulBuild/artifact/ui/webapp/target/cluwe.webapp-<version>.war</pre>
14.	Copy the contents of the deploy/prd folder from the release branch of SVN to the /home/cluwe_services/install directory
15.	<p>Execute the following two commands to update the service management scripts with the correct version of jar file to start. Substitute the release version for the <version> string.</p> <pre>sed -i -- 's/ARTIFACT_VERSION="[:digit:][:digit:][:digit:]" /ARTIFACT_VERSION="<version>"/' *.sh</pre>
16.	Verify that step 6 was successful by ensuring that the ARTIFACT_VERSION value in locker.sh matches the version in the locker.adapter-<version>.jar file.
17.	<p>Copy the jar files and shell scripts to their installation directories using the commands below:</p> <pre>cp ~/install/locker* /opt/cluwe/locker.adapter</pre> <pre>cp ~/install/provision* /opt/cluwe/provision</pre> <pre>cp ~/install/signature* /opt/cluwe/signature.adapter</pre> <pre>cp ~/install/version* /opt/cluwe/version.adapter</pre> <pre>cp ~/install/computejob* /opt/cluwe/computejob</pre> <pre>cp ~/install/bjobs.sh /opt/cluwe/computejob</pre> <pre>cp ~/install/submitsasjob.sh /opt/cluwe/sasgsub</pre>
18.	Use the process <i>Start the CLUWE Services</i> process in this document to start the processes on their servers.

Once the service(s) file is deployed, verification should be performed using the Verification Process in the *CLUWE Disaster Recovery Plan*. If deploying a new webapp and new services, the verification only need to be performed once at the end of all the deployments.

7.4 PRD Deployment

7.4.1 Build the Release Code

The same code used in the last QA deploy is also used for the PRD deploy so a new build is not. The tag that contains the last code deployed to QA must be tagged using the name PRD-<Maven Version>-<CHG Number> and the contents of that tag stored in TFS under a directory of the same name.

7.4.2 Webapp Deployment

This process wraps steps around the WHO deployment process to archive the code and deployed war file before deployment.

Step	Details
1	<p>Create a ServiceNow request to deploy a WebSite using the form <i>Home > Order Things > Technology Infrastructure > Web Hosting and Analytics > Website/Application Deployment</i></p> <p>Use the following values when filling out the form:</p> <p>Select appropriate platform: JBOSS</p> <p>Which environment are you deploying to? PRD</p> <p>Specify Application Name: CLUWE</p> <p>Is this the first time this code has ever been deployed? No</p> <p>What server are you deploying FROM? cluwe-d1</p> <p>What server are you deploying TO? cluwe-p1</p> <p>What file(s) and/or directories require deployment (please include full path)? <Same path as the last QA deploy></p> <p>Where do you want the file or folders copied TO (please include full path)? /od/cluwe/installableApp/cluwe.war</p>
2	<p>Once the new war file is deployed, verification should be performed using the Verification Process in the <i>CLUWE Disaster Recovery Plan</i>. If deploying a new webapp and new services, the verification only need to be performed once at the end of all the deployments.</p>

7.4.3 Service Deployment

Step	Details
1.	SSH to cluwe-p3.am.lilly.com then execute the command “ <code>sudo /etc/bcluwe_services</code> ” without the quotes. When prompted, enter your Lilly password.
2.	Stop any running instances of the service you’re going to deploy using the <i>Stop the CLUWE Services</i> process in this document.
3.	Change to the /home/cluwe_services/install directory and remove any existing files.
4.	Transfer the contents of the /home/cluwe_services_q/install directory on cluwe-q3.am.lilly.com to /home/cluwe_services/install on cluwe-p3.am.lilly.com.
5.	<p>Copy the jar files and shell scripts to their installation directories using the commands below:</p> <pre>cp ~/install/locker* /opt/cluwe/locker.adapter cp ~/install/provision* /opt/cluwe/provision cp ~/install/signature* /opt/cluwe/signature.adapter cp ~/install/version* /opt/cluwe/version.adapter cp ~/install/computejob* /opt/cluwe/computejob</pre>

Step	Details
	<pre>cp ~/install/bjobs.sh /opt/cluwe/computejob</pre> <pre>cp ~/install/submitsasjob.sh /opt/cluwe/sasgsub</pre>
6.	Use the process <i>Start the CLUWE Services</i> process in this document to start the processes on their servers.

Once the service(s) file is deployed, verification should be performed using the Verification Process in the *CLUWE Disaster Recovery Plan*. If deploying a new webapp and new services, the verification only need to be performed once at the end of all the deployments.

8. PERIODIC REVIEW

Periodic reviews for this system will follow frequency and content guidelines as defined in the *ITC-SOP-IT Periodic Review-8494*.

9. SYSTEM INVENTORY

The CMDB in ServiceNow is the inventory tool used to manage this system.

10. INFRASTRUCTURE AND INTERFACE SUPPORT INFORMATION

10.1 Infrastructure Support

Service	Support Team	Service Now Group
Isilon NAS	Storage Team	PLATFORM-NAS-GLB
Redmine SVN	High Performance Computing Team	PLATFORM-HPC-Z1
Team Foundation Server (TFS)	Global SEST Team	ITTOOLS-TFS-GAIT-IBM-GLB
jBoss Web Servers and associated F5 Load Balancer	Web Hosting Team	WHO-GLB
ActiveMQ Brokers	Web Hosting Team	WHO-GLB
Oracle Database	Global Data Management	DATABASE-ORACLE-GLB

10.2 Interface Support

Service	Support Team	Service Now Group
SAS Grid Job Scheduling (sasgsub)	Enterprise Analytics	ANALYTICS-EA-GLB
SDD Integration Broker	Global Stats	GSS-LRLIT-GLB
Isilon Web Services and associated F5 Load Balancer (Isilon Platform API)	Storage Team	PLATFORM-NAS-GLB

Service	Support Team	Service Now Group
Taffy	Global Stats	GSS-LRLIT-GLB
Clinical Data Transfer Scheduler (CDTS)	Global Stats	GSS-LRLIT-GLB
Email	Messaging Services	HP-GLB-MSG-O365-L3
Text	Mobile Messaging Solutions, Inc. (MMS) 1-800-290-5141	MMS does not have a Service Now Group

10.3 Necessary Support Access

The onboarding process for CLUWE IT SMEs is defined in the *CLUWE Web Tool Security Administration SOP*. This process includes requesting the necessary support access.

10.4 Necessary Testing Access

The onboarding process for CLUWE IT SMEs is defined in the *CLUWE Web Tool Security Administration SOP*. This process includes requesting the necessary support access.

10.5 Necessary Development Access

The onboarding process for CLUWE IT SMEs is defined in the *CLUWE Web Tool Security Administration SOP*. This process includes requesting the necessary support access.

11. ROUTINE ADMINISTRATION TASKS

Routine administrative tasks are activities that are necessary for the ongoing maintenance and do not represent a change to software or hardware of the system.

11.1 Add new Electronic Signature meaning entry

To add a new entry, execute the following two SQL statements against the dev, qa or prd database:

```
insert into cluwe_owner.esig_meaning (sig_meaning, transaction_id) values ('<new sig meaning>', '<CHG #>');
commit;
```

Where <new sig meaning> is the text for the new entry and <CHG #> is the id of the Service Now Change created for the change. The change takes effect immediately.

11.2 Adjust the number of scheduled sas jobs displayed to the user

The property below adjusts how many SAS Grid jobs are displayed in the Schedule tab by only displaying jobs newer than this value. The values set at initial deployment were set based on performance testing. Adjusting the values requires that the application.properties file must be modified and the war file that contains it must be redeployed.

Property	Details
scheduler.maxDaysToDisplay	Don't display SAS jobs older than this value in days.

11.3 Display overall version, signature and file lock statistics

Execute the following SQL statement against the dev, qa or prd database:

```
with tid as (
    select from_tz(APPROVAL_TIMESTAMP_UTC, 'UTC') at time zone
    'America/New_York' as utc, transaction_id, 'esig' as type, 'SUCCESS' as
    status from CLUWE_OWNER.ESIG_LOG

    UNION ALL

    select from_tz(locked_utc, 'UTC') at time zone 'America/New_York' as utc,
    transaction_id, 'lock' as type, status from CLUWE_OWNER.FILELOCK_LOG

    UNION ALL

    select from_tz(versioned_utc, 'UTC') at time zone 'America/New_York' as
    utc, transaction_id, 'version' as type, status from CLUWE_OWNER.VERSION_LOG
)
select
    sum(CASE WHEN tid.type = 'esig' THEN 1 ELSE 0 END) as "eSig Success",
    sum(CASE WHEN tid.type = 'lock' and status = 'SUCCESS' THEN 1 ELSE 0 END)
as "Lock Success",
    sum(CASE WHEN tid.type = 'lock' and (status != 'SUCCESS' or status is
    null) THEN 1 ELSE 0 END) as "eSig Failed",
    sum(CASE WHEN tid.type = 'version' and status = 'SUCCESS' THEN 1 ELSE 0
    END) as "Version Success",
    sum(CASE WHEN tid.type = 'version' and (status != 'SUCCESS' or status is
    null) THEN 1 ELSE 0 END) as "Version Failed"
from tid
order by 1 desc
;
```

11.4 Scan the database for failed version operations

Execute the following SQL statement against the dev, qa or prd database:

```
select * from cluwe_owner.version_log where status != 'SUCCESS' or status is
null order by versioned_utc;
```

When copying and pasting the above statement the single quotes around the word SUCCESS must be replaced with regular single quotes as Microsoft Word converts them to illegal characters. The results can include errors that occurred due to user error so the value of the status column must be considered.

11.5 Scan the database for failed Lock-in-Place operations

Execute the following SQL statement against the dev, qa or prd database:

```
select * from cluwe_owner.filelock_log where status != 'SUCCESS' or status is
null order by locked_utc desc;
```

When copying and pasting the above statement the single quotes around the word SUCCESS must be replaced with regular single quotes as Microsoft Word converts them to illegal characters. The results can include errors that occurred due to user error so the value of the status column must be considered.

11.6 Scan the database for failed Signature operations

Execute the following SQL statement against the dev, qa or prd database:

```
select * from cluwe_owner.ESIG_LOG where STATUS != 'SUCCESS' or STATUS is  
null order by APPROVAL_TIMESTAMP_UTC desc;
```

When copying and pasting the above statement the single quotes around the word SUCCESS must be replaced with regular single quotes as Microsoft Word converts them to illegal characters. The results can include errors that occurred due to user error so the value of the status column must be considered.

11.7 Verify that Lock-In-Place is operating

Execute the following SQL statement against the dev, qa or prd database:

```
select * from CLUWE_OWNER.JOB_INFO_LOG order by FIRE_TIME desc;
```

The top row will detail the operation of the last Lock-In-Place execution. All timestamps are in UTC:

- BEGIN_WINDOW: Files older than this timestamp will not be considered for locking.
- END_WINDOW: Files newer than this timestamp will not be considered for locking.
- FIRE_TIME: The timestamp of when the job started executing.
- JOB_NAME: This column has multiple values embedded:
 - "FileLockingJob": The name of the scheduler job.
 - (X/Y): X is the number of files eligible for locking. Y is the maximum number of files that will be locked in the execution.
 - TransactionId: The CLUWE transaction id assigned when the job started.
- RUN_TIME: The time in milliseconds it took the job to execute.

The FIRE_TIME value shows the last time the job started. If the FIRE_TIME value is older than 20 minutes then the job is not operating properly.

12. SERVICE MANAGEMENT TASKS

12.1 Start the CLUWE Services

The CLUWE services are java applications that run on virtual RedHat Linux servers where each server is running an instance of each service. The services execute under a service account so you must sudo into that account to manage them.

The service names, their installation directories and management shell scripts are listed in the section named Deployable Artifact Information within this document. Use this section to determine which servers each service should be started on.

Step	Action
1	SSH into the CLUWE Services server using your Lilly user id and password
2	Execute the command "sudo /etc/<sudo account>" without the quotes where <sudo account> is: <ul style="list-style-type: none">• bcluwe_services for production• bcluwe_services_q for QA• bcluwe_services_d for dev Then enter your Lilly password when prompted.
3	Change to the installation directory for the service.

4	A shell script named after the service exists in each directory and this script is used to control the service. For each service, enter the shell script name followed by “start”. Enter the command that corresponds to the service below. You should see a result from both commands within 90 seconds. If a new version of the service has just been deployed then the start command may either fail or not return for several minutes. If this occurs you can press ctrl-c to stop the script then verify that the process is running using the instructions within this document.
5	Verify the service is running using the section of this document named “Verify that CLUWE Services are running”.
6	This process must be repeated on each of the Executing Servers for the service to start all the instances.

Each of the service directories also contains the log files for all the running service instances. The log files are named with the name of the service and the server it’s running on.

12.2 Start the CLUWE Services in Debug Mode

To assist in troubleshooting problems, the CLUWE services can be started with increased logging enabled. This process doesn’t impact any functionality but could add a small amount of processing overhead. It is not recommended to run a service in debug mode unless troubleshooting an issue as debug generates a large amount of log entries making it difficult to find warning and error messages.

The service names, their installation directories and management shell scripts are listed in the section named Deployable Artifact Information within this document. Use this section to determine which servers each service should be started on.

Step	Action
1	SSH into the CLUWE Services server using your Lilly user id and password
2	Execute the command “sudo /etc/<sudo account>” without the quotes where <sudo account> is: <ul style="list-style-type: none"> • bcluwe_services for production • bcluwe_services_q for QA • bcluwe_services_d for dev Then enter your Lilly password when prompted.
3	Change to the installation directory for the service to start
4	A shell script named after the service exists in each directory and this script is used to control the service. For each service, enter the shell script name followed by “start debug”. Enter the command that corresponds to the service below. You should see a result from both commands within 90 seconds. If a new version of the service has just been deployed then the start command may either fail or not return for several minutes. If this occurs you can press ctrl-c to stop the script then verify that the process is running using the instructions within this document.
5	Verify the service is running using the section of this document named “Verify that CLUWE Services are running”.
6	This process must be repeated on each of the Executing Servers for the service to start all the instances.

Each of the service directories also contains the log files for all the running service instances. The log files are named with the name of the service and the server it's running on.

12.3 Stop the CLUWE Services

Step	Action
1	SSH into the CLUWE Services server using your Lilly user id and password
2	<p>Execute the command "sudo /etc/<sudo account>" without the quotes where <sudo account> is:</p> <ul style="list-style-type: none"> • bcluwe_services for production • bcluwe_services_q for QA • bcluwe_services_d for dev <p>Then enter your Lilly password when prompted.</p>
3	Change to the installation directory for the service to start
4	<p>A shell script named after the service exists in each directory and this script is used to control the service. For each service, enter the shell script name followed by "stop". Enter the command that corresponds to the service below. You should see a result from both commands within 90 seconds.</p> <p>If a new version of the service has just been deployed then the start command may either fail or not return for several minutes. If this occurs you can press ctrl-c to stop the script then verify that the process is running using the instructions within this document.</p>
5	<p>Verify the service stopped by executing the following command:</p> <pre>ps -ef grep spring.profiles</pre> <p>The command will list the running processes so the stopped process will not be listed.</p>
6	This process must be repeated on each of the Executing Servers for the service to start all the instances.

Each of the service directories also contains the log files for all the running service instances. The log files are named with the name of the service and the server it's running on.

12.4 Version Service Monitoring

The status of the Versioning service can be obtained by visiting the following web page:

<https://cluwe.am.lilly.com/monitor/version>

An HTTP 200 message will be returned if one or more instances of the service are running, otherwise, an HTTP 500 message will be returned indicating that no instances of the services are running. A web service client or Chrome must be used to access this URL as Internet Explorer will not display the HTTP status.

12.5 Signature Service Monitoring

The status of the Signature service can be obtained by visiting the following web page:

<https://cluwe.am.lilly.com/monitor/signature>

An HTTP 200 message will be returned if one or more instances of the service are running, otherwise, an HTTP 500 message will be returned indicating that no instances of the services are running. A web service client or Chrome must be used to access this URL as Internet Explorer will not display the HTTP status.

12.6 Verify that CLUWE Services are running on a server

The CLUWE Services are java applications running on Linux servers so verification that they're running is done the same way as any other Linux application. SSH into each of the CLUWE Services linux servers using your Lilly user id and password. Once logged in execute the following command without quotes "`ps -ef | grep spring.profiles`" and you should see several lines returned. Each line should contain the text below where `<service name>` is the name of the service and `version` is the currently running version.

```
/home/cluwe_services/java/java_home/bin/java -jar -  
Dspring.profiles.active=prd <service name>-<version>.jar
```

If the service isn't listed, it is not running. The valid list of services is:

- version.adapter
- signature.adapter
- provision
- locker.adapter
- computeJob

13. REFERENCES

A current list of system-specific documents stored in Regulus can be generated using Regulus. The documents are stored in the IT Library Regulus repository at IT_Library > LRL IT > Systems C to D > CLUWE (Clinical User Working Environment) and IT_Library > LRL IT Secure > Systems C to D > CLUWE (Clinical User Working Environment).