**Portal Server**

**User Guide**

Version 0.1

© Sterlite Technologies Ltd.

**CONFIDENTIALITY CLAUSE**

No part of this document may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, recording, photocopying or otherwise without the prior written permission of Sterlite Technologies Ltd.

The contents of this document are provided to **“DMC”** in confidence solely for the purpose of evaluating possible business relationship.

**ALL RIGHTS RESERVED**

Sterlite Technologies Ltd.

Elitecore, Block 6, Magnet Corporate Park,

Nr. Sola Flyover, Thaltej

Ahmedabad–380059, India

**TRADEMARKS**

All the brand names and other products or services mentioned in this document are identified by the trademarks or service marks of their respective owners.

**DISCLAIMER**

The document is designed as per the scope of DMC., contents in this document are mutually agreed by Sterlite Tech & DMC. This document (Solution Requirement Specifications) Supersede all the previously shared documents (SOW & Previous SRS versions). SRS (Solution Requirement Specifications) is the base of development & configurations. Anything over & above this signed off SRS shall be mutually discussed and agreed keeping DMC business mission accomplishment as priority, any significant additions in scope shall be mutually discussed and handled separately as a change request. New change request should not impact on the delivery & commercial milestones of this project.

| Point of Contact | | | |
| --- | --- | --- | --- |
| Name |  |  |  |
| Title |  |  |  |
| Email |  | Skype Id |  |
| Mobile |  |  |  |

|  |  |
| --- | --- |
| Target Audience | |
| 1 | CEIR Agency |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| Reference& Resources | | Remark |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| Open Points Tracker | | Remark |
|  |  |  |
|  |  |  |
|  |  |  |

**VERSION CONTROL AND DOCUMENT HISTORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **Document Version** | **Date** | **Owner** | **Document Status** | **Remarks** |
| 1 | 0.1 |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table of Contents

1 Overview 6

1.1 User Request Process Overview 7

1.2 User Request – Detailed Description 7

2 Architecture 7

2.1 Physical Architecture Overview 8

2.2 Software Architecture Overview 8

2.2.1 SLAModule.jar 8

2.2.2 BlackListProcess.jar 8

2.2.3 GreyListProcess.jar 9

2.2.4 BlackListProcess.jar 9

2.2.5 CEIRCdrParser.jar 9

3 Process Overview 10

3.1 Process Overview 10

3.2 Directory Structure 10

4 Process Overview - Description 11

4.1 Tomcat 11

4.2 GUI 11

4.3 APIService1.jar 11

4.4 APIService2.jar 12

4.5 APIService3.jar 13

4.6 APIService4.jar 14

4.7 BlackListFileProcess.jar 15

4.8 BlackListProcess.jar 16

4.9 GreyListProcess.jar 17

4.10 CEIRCdrParser.jar 18

4.11 SLAModule.jar 18

5 Process Management 19

5.1 Starting the Process 19

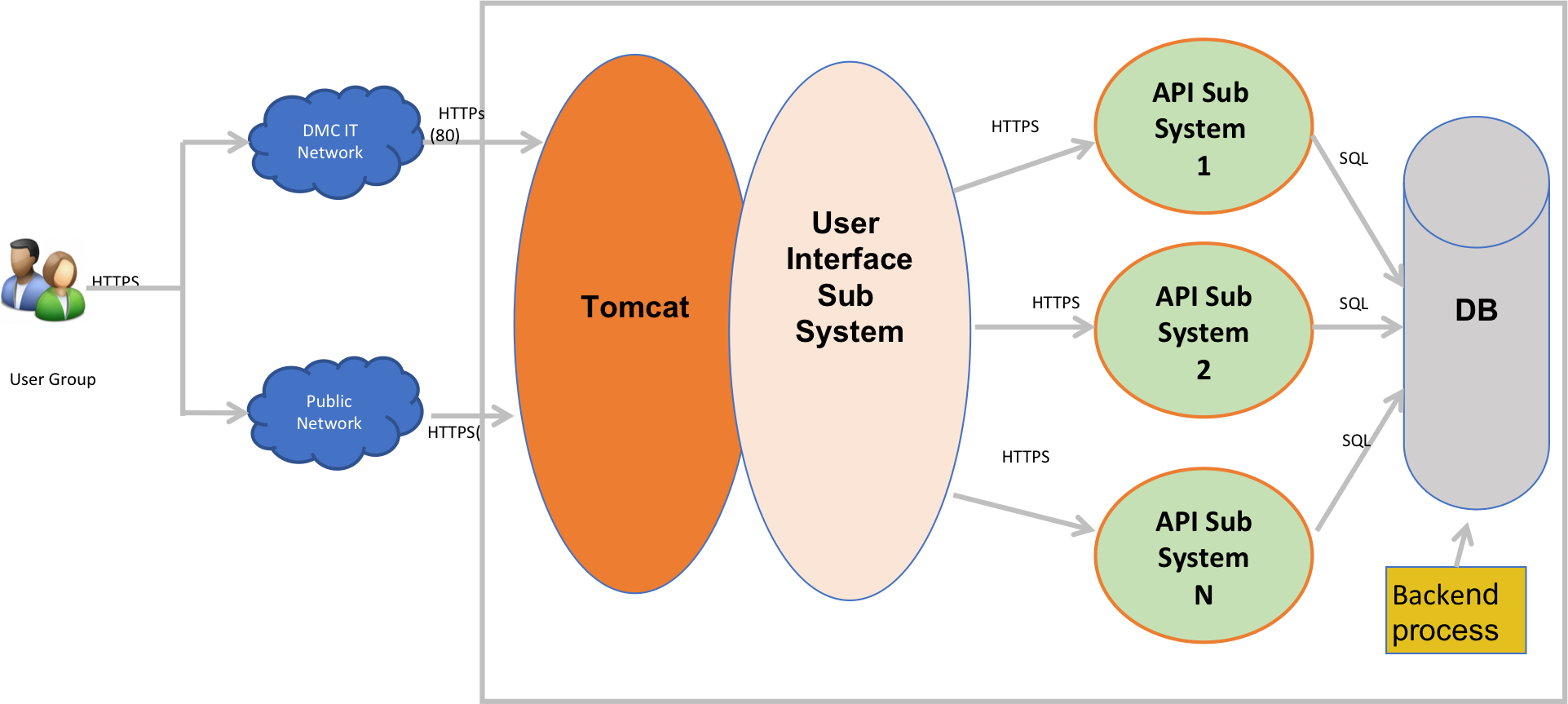
5.2 Stopping the Process 20

5.3 Checking the Status of the Process 20

# Overview

Portal Server subsystem is one of the key component of the CEIR system. This subsystem host the software processes that does the following

1. Host User portal for both external and internal set of users
2. Process the request as raised by users and build the inactive device Database



The figure above depicts the flow. The flow is as follows

* User send the request to the CEIR system over Web interface
* System process the request and update Database

The key components of the portal server are as follows:

1. Tomcat – This is the web server application which implement HTTP server functionality and host the GUI front end application
2. User Interface sub system – This is deployed on tomcat as war file. This host all the front end view that is displayed on the browser. For any interaction with database, the user interface subsystem calls API of API sub system to perform operations on the database.
3. API Sub system – There are mutltple instances of this subsystem. This subsystem implement API which interact with database
4. Backend Process are system process that work on the user request and process and update the database. For example, user raise a consignment which is marked in NEW state when it is submitted by user. The backend sub system process request in NEW state and apply the rules and accordingly update the status of the request in the database
5. Database is a external component which his hosted on the database server

## User Request Process Overview

The key process on the portal server is the different user request as received by the user for updating the inactive Device Database. Some of the feature request raised by different external users are as follows:

* Consignment
* Stock
* Manage Type Approval
* Grievance

All web portal request raised by user are fulfilled by this portal server

A typical flow for a request is as follows:

* User login into the portal
* User select a request say, importer want to raise the consignment
* User raise a consignment and submit the request to the server
* Server receives the request on the GUI module hosted on the web server
* The GUI module call the API on the respective API subsystem for further processing
* API subsystem interact with Database for further processing and send response back to the GUI sub system
* GUI subsystem receive the response, reformat the response as per browser and send to the user.

## User Request – Detailed Description

All the process like tomcat, API subsystems etc. are always running.

These processes are running using system crontab task.

There is a monitoring script also running which ensure all such processes are running.

All API subsystem and GUI subsystem are multi-threaded, so multiple requests can be processed at same time.

All the user requests are processed in the offline mode. Once the request is received and updated in the DB, there is a system backend process, which run and process all the user request in a sequential manner. This mean that only one request will be processed at one time.

The backend process poll for the oldest request in the database with status marked as NEW. All such requests are processed and if found ok, the status is changed to PENDING\_FOR\_CEIR\_ADMIN\_APPROVAL

# Architecture

This section describes the physical and software architecture of the Portal server

## Physical Architecture Overview

There are 2 physical server working in active standby mode. Both the server share a virtual IP.

The URL is mapped to the virtual IP which is hosted on one of the servers.

## Software Architecture Overview

All the software processes are deployed on both the servers. The software deployed in the following fashion:

|  |  |  |
| --- | --- | --- |
| Process | Mode | Remarks |
| Tomcat | Active-Active | It runs on both server waiting for the request to come from user |
| GUI sub system | Active- Active | This run on the tomcat, so the same mode as applicable for tomcat is applicable. |
| API sub system | Active-Active | It runs on both server waiting for the request to come from GUI subsystem |
| Backend Subsystem | Active-Standby | This process run on the server having the virtual IP |

There are multiple backend process running that are configured in active-standby mode.

There are the following 5 backend process. The software architecture for each component is as follows

### SLAModule.jar

This process generates the SLA report. All the user generated request, if found to be ok, need approval from the CEIR admin. There is an SLA for each request type. For example, CEIR admin has to approve the consignment request within 3 days. In case such request are still pending for approval on 4th day, the entry would be created in the SLA report.

### BlackListProcess.jar

This process creates the blacklist database from grey list database.

Once the device is marked as greylisted, and the timer period has expired, then the devices should be moved to black list. The timer period depends on the blocking type.

A day end process run which move the entry from greylist table to black list table.

### GreyListProcess.jar

This process creates the greylist file from grey list database.

When any device is marked as stolen or blocked, then such devices are added in the grey list table in the system. However such records are to be shared with operator.

A day end process run which create the greylist file. Two kind of files are created. One is full and other is incremental.

The file type as “Full” indicate the current snapshot of the greylist table.

The file type as “incremental” indicate the addition/deletion in the greylist table for that given day.

Such file can be downloaded from the “Grey List” section for the Operator CEIR Portal.

### BlackListProcess.jar

This process creates the blacklist file from black list database.

A day end process run which create the blacklist file. Two kind of files are created. One is full and other is incremental.

The file type as “Full” indicate the current snapshot of the blacklist table.

The file type as “incremental” indicate the addition/deletion in the blacklist table for that given day.

Such file can be downloaded from the “Black List” section for the Operator CEIR Portal.

### CEIRCdrParser.jar

This process processes the user request and update the inactive database as per the request type.

This process runs every minute to check if there are new request. The process is scheduled to run from crontab.

If there are new request, the process execute the oldest request in a sequential manner.

Based on the processing, there are 2 outcomes: either the request would move forward and go to CEIR admin for approval or the request would be sent back to user for correction.

# Process Overview

This section describe the process details deployed on portal server. Each section describes the process, related configuration files, log files

## Process Overview

The following processes are deployed on portal server

|  |  |  |
| --- | --- | --- |
| SN | Process Name | Description |
| 1 | Tomcat | Open source HTTP(s) based web server |
| 2 | GUI | Running in tomcat container process |
| 3 | APIService1.jar | One of the API subsystem hosting the APIs |
| 4 | APIService2.jar | One of the API subsystem hosting the APIs |
| 5 | APIService3.jar | One of the API subsystem hosting the APIs |
| 6 | APIService4.jar | One of the API subsystem hosting the APIs |

All these processes are running from the crontab entry. In case the process is not running then the process is started.

The following process run on a predefined frequency. These are mostly back end processes.

|  |  |  |
| --- | --- | --- |
| SN | Process Name | Description |
| 1 | SLAModule.jar | This process generate the SLA report. |
| 2 | BlackListProcess-.jar | This process creates the blacklist database from grey list database. |
| 3 | GreyListProcess.jar | This process creates the greylist file from grey list database. |
| 4 | BlackListFileProcess.jar | This process creates the blacklist file from black list database. |
| 5 | CEIRCdrParser.jar | This process process the user request and update the inactive database as per the request type |

## Directory Structure

The section describe the key folder where the software are deployed in the system

|  |  |  |
| --- | --- | --- |
| SN | Description | Value |
| 1 | Base Path where the software is deployed | /u01/ceirapp  There is a folder for each subsystem/module |
| 2 | Base path where the logs are maintained | /u02/ceirdata/  There is a folder for each subsystem/module |
| 3 | Base Path where the file is saved uploaded by user/ export file/ system generated file Eg. Stock file | /u01/ceirapp/apache-tomcat-9.0.34/webapps/docs/ceir/ |

# Process Overview - Description

This section description the process, the configuration file and log files.

## Tomcat

Tomcat is third party open source web server software application. The tomcat is opened on port 80 to receive the request

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/ apache-tomcat-9.0.34/bin |
| 2 | Log File Path | /u01/ceirapp/ apache-tomcat-9.0.34/logs |
| 3 | Configuration File Path | /u01/ceirapp/apache-tomcat-9.0.34/conf |
| 4 | Script to start/stop tomcat | /u01/ceirapp/Tomcat |

Logs are created day wise in the same folder

## GUI

GUI is frontend software application. The tomcat is opened on port 80 to receive the request

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/ apache-tomcat-9.0.34/webapps |
| 2 | Log File Path | /u02/ceirdata/GUI |
| 3 | Configuration File Path | /u01/ceirapp/GUI |

The configuration File (application.properties) has internal parameter and nothing to be changed.

The messages.properties and messages\_km.propertis are file for support multiple languages.

Logs are created day wise in the same folder

## APIService1.jar

This API subsystem process and handles the following requests

<< add that table >>

This subsystem is listening to request on port no 9503

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/APIService1 |
| 2 | Log File Path | /u02/ceirdata/APIService1 |
| 3 | Configuration File Path | /u01/ceirapp/APIService1 |

The configuration File (application.properties) description is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | spring.datasource.url | DB URL which specifies DB name and port | jdbc:oracle:thin:@dmc-prod-db:1521/dmcproddb |
| 2 | spring.datasource.username | DB username | CRESTELCEIR |
| 3 | spring.datasource.password | DB password | CRESTELCEIR |
| 4 | Local-ip | Physical IP address of the server | 172.24.2.57 |
| The file contains other configuration parameter which should not be changed at any point of time | | | |

The logback.xml file is set as follow:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | FileNamePattern | Log Path along with file name |  |

Rest of the parameter should not be changed in the file.

Logs are created day wise in the same folder.

There are 3 scripts present for the following purpose

|  |  |  |
| --- | --- | --- |
| SN | Script | Description |
| 1 | Start.sh | This script is used to start the process |
| 2 | Stop.sh | This script is used to stop the process |
| 3 | Monitor.sh | This script is used to check if the process is running or not. |

## APIService2.jar

This API subsystem process and handles the following requests

<< add that table >>

This subsystem is listening to request on port no 9502

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/APIService2 |
| 2 | Log File Path | /u02/ceirdata/APIService2 |
| 3 | Configuration File Path | /u01/ceirapp/APIService2 |

The configuration File (application.properties) description is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | spring.datasource.url | DB URL which specifies DB name and port | jdbc:oracle:thin:@dmc-prod-db:1521/dmcproddb |
| 2 | spring.datasource.username | DB username | CRESTELCEIR |
| 3 | spring.datasource.password | DB password | CRESTELCEIR |
| 4 | Local-ip | Physical IP address of the server | 172.24.2.57 |
| The file contains other configuration parameter which should not be changed at any point of time | | | |

The logback.xml file is set as follow:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | FileNamePattern | Log Path along with file name |  |

Rest of the parameter should not be changed in the file.

Logs are created day wise in the same folder.

There are 3 scripts present for the following purpose

|  |  |  |
| --- | --- | --- |
| SN | Script | Description |
| 1 | Start.sh | This script is used to start the process |
| 2 | Stop.sh | This script is used to stop the process |
| 3 | Monitor.sh | This script is used to check if the process is running or not. |

## APIService3.jar

This API subsystem process and handles the following requests

<< add that table >>

This subsystem is listening to request on port no 9504

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/APIService3 |
| 2 | Log File Path | /u02/ceirdata/APIService3 |
| 3 | Configuration File Path | /u01/ceirapp/APIService3 |

The configuration File (application.properties) description is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | spring.datasource.url | DB URL which specifies DB name and port | jdbc:oracle:thin:@dmc-prod-db:1521/dmcproddb |
| 2 | spring.datasource.username | DB username | CRESTELCEIR |
| 3 | spring.datasource.password | DB password | CRESTELCEIR |
| 4 | Local-ip | Physical IP address of the server | 172.24.2.57 |
| The file contains other configuration parameter which should not be changed at any point of time | | | |

The log4j.properties file is set as follow:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | Log4j.appender.stdout.File | Log Path along with file name |  |

Rest of the parameter should not be changed in the file.

Logs are created day wise in the same folder.

There are 3 scripts present for the following purpose

|  |  |  |
| --- | --- | --- |
| SN | Script | Description |
| 1 | Start.sh | This script is used to start the process |
| 2 | Stop.sh | This script is used to stop the process |
| 3 | Monitor.sh | This script is used to check if the process is running or not. |

## APIService4.jar

This API subsystem process and handles the following requests

<< add that table >>

This subsystem is listening to request on port no 9501

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/APIService4 |
| 2 | Log File Path | /u02/ceirdata/APIService4 |
| 3 | Configuration File Path | /u01/ceirapp/APIService4 |

The configuration File (application.properties) description is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | spring.datasource.url | DB URL which specifies DB name and port | jdbc:oracle:thin:@dmc-prod-db:1521/dmcproddb |
| 2 | spring.datasource.username | DB username | CRESTELCEIR |
| 3 | spring.datasource.password | DB password | CRESTELCEIR |
| 4 | Local-ip | Physical IP address of the server | 172.24.2.57 |
| The file contains other configuration parameter which should not be changed at any point of time | | | |

The logback.xml file is set as follow:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | FileNamePattern | Log Path along with file name |  |

Rest of the parameter should not be changed in the file.

Logs are created day wise in the same folder.

There are 3 scripts present for the following purpose

|  |  |  |
| --- | --- | --- |
| SN | Script | Description |
| 1 | Start.sh | This script is used to start the process |
| 2 | Stop.sh | This script is used to stop the process |
| 3 | Monitor.sh | This script is used to check if the process is running or not. |

## BlackListFileProcess.jar

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/BackendProcess/BlackListFileProcess |
| 2 | Log File Path | /u02/ceirdata/ BackendProcess/ BlackListFileProcess |
| 3 | Configuration File Path | / u01/ceirapp/BackendProcess/ BlackListFileProcess |

The configuration File (application.properties) description is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | spring.datasource.url | DB URL which specifies DB name and port | jdbc:oracle:thin:@dmc-prod-db:1521/dmcproddb |
| 2 | spring.datasource.username | DB username | CRESTELCEIR |
| 3 | spring.datasource.password | DB password | CRESTELCEIR |
| The file contains other configuration parameter which should not be changed at any point of time | | | |

The log4.properties file is set as follow:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | Log4j.appender.A.File | Log Path along with file name |  |

Rest of the parameter should not be changed in the file.

Logs are created day wise in the same folder.

## BlackListProcess.jar

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/BackendProcess/BlackList |
| 2 | Log File Path | /u02/ceirdata/ BackendProcess/ BlackList |
| 3 | Configuration File Path | / u01/ceirapp/BackendProcess/ BlackList |

The configuration File (application.properties) description is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | spring.datasource.url | DB URL which specifies DB name and port | jdbc:oracle:thin:@dmc-prod-db:1521/dmcproddb |
| 2 | spring.datasource.username | DB username | CRESTELCEIR |
| 3 | spring.datasource.password | DB password | CRESTELCEIR |
| The file contains other configuration parameter which should not be changed at any point of time | | | |

The log4.properties file is set as follow:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | Log4j.appender.A.File | Log Path along with file name |  |

Rest of the parameter should not be changed in the file.

Logs are created day wise in the same folder.

## GreyListProcess.jar

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/BackendProcess/GreyList |
| 2 | Log File Path | /u02/ceirdata/ BackendProcess/ GreyList |
| 3 | Configuration File Path | / u01/ceirapp/BackendProcess/ GreyList |

The configuration File (application.properties) description is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | spring.datasource.url | DB URL which specifies DB name and port | jdbc:oracle:thin:@dmc-prod-db:1521/dmcproddb |
| 2 | spring.datasource.username | DB username | CRESTELCEIR |
| 3 | spring.datasource.password | DB password | CRESTELCEIR |
| The file contains other configuration parameter which should not be changed at any point of time | | | |

The log4.properties file is set as follow:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | Log4j.appender.A.File | Log Path along with file name |  |

Rest of the parameter should not be changed in the file.

Logs are created day wise in the same folder.

## CEIRCdrParser.jar

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/BackendProcess/CeirParser |
| 2 | Log File Path | /u02/ceirdata/ BackendProcess/CeirParser |
| 3 | Configuration File Path | / u01/ceirapp/BackendProcess/CeirParser/conf |

The configuration File (config.properties) description is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | db.url | DB URL which specifies DB name and port | jdbc:oracle:thin:@dmc-prod-db:1521/dmcproddb |
| 2 | username | DB username | CRESTELCEIR |
| 3 | password | DB password | CRESTELCEIR |
| 4 | Local IP | Local IP of the machine | 172.24.2.57 |
| The file contains other configuration parameter which should not be changed at any point of time | | | |

The log4.properties l file is set as follow:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | Log4j.appender.A.File | Log Path along with file name |  |

Rest of the parameter should not be changed in the file.

Logs are created day wise in the same folder.

## SLAModule.jar

The directory structure details are as follows:

|  |  |  |
| --- | --- | --- |
| SN | Task | Description |
| 1 | Deployment Path | /u01/ceirapp/BackendProcess/SLA |
| 2 | Log File Path | /u02/ceirdata/BackendProcess/SLA |
| 3 | Configuration File Path | / u01/ceirapp/BackendProcess/SLA |

The configuration File (application.properties) description is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | spring.datasource.url | DB URL which specifies DB name and port | jdbc:oracle:thin:@dmc-prod-db:1521/dmcproddb |
| 2 | spring.datasource.username | DB username | CRESTELCEIR |
| 3 | spring.datasource.password | DB password | CRESTELCEIR |
| The file contains other configuration parameter which should not be changed at any point of time | | | |

The log4.properties l file is set as follow:

|  |  |  |  |
| --- | --- | --- | --- |
| SN | Task | Description | Sample value |
| 1 | Log4j.appender.A.File | Log Path along with file name |  |

Rest of the parameter should not be changed in the file.

Logs are created day wise in the same folder.

# Process Management

This section describes how the process need to be managed. This includes how to start/stop the application and check the status of the application

For all process, the steps to start and stop the application is same

## Starting the Process

Kindly perform the following steps

1. Login to the server
2. Go to the software installation path
3. Run the following command
   1. ./startup.sh
4. The following output should be displayed which mean that the process has been succesfully started.
   1. Process Started
   2. If the process is already running, the output will show the following
      1. Process already running

## Stopping the Process

Kindly perform the following steps

1. Login to the server
2. Go to the software installation path
3. Run the following command
   1. ./stop.sh
4. The following output should be displayed which mean that the process has been succesfully stopped.
   1. Process Successfully Stopped
   2. If the process is not running, the output will show the following
      1. No Process running

## Checking the Status of the Process

Kindly perform the following steps

1. Login to the server
2. Go to the software installation path
3. Run the following command
   1. ./monitor.sh
4. The following output should be displayed which mean that the process is running.
   1. Process Running
   2. If the process is not running, the output will show the following
      1. No Process Running

