IIP Installation and configuration Manual

Version 1.2.0, Mar 23 2024: Updated version for IIP release 1.4.0

Table of Contents

Chapter 1. Introduction

This is a document with instructions on how to install and configure the IUCLID Integration Platform (IIP), a software provided by CropLife Europe.

1.1. Version history

Date	Version	Comment
Feb 01 2023	1.0	First release, for the IIP version 1.2.0
Feb 02 2023	1.0.1	Small enhancements & error corrections
Jun 11 2023	1.1.0	Added new config parameters for the IIP version 1.3.0
Mar 23 2024	1.2.0	Added new config parameters for the IIP version 1.4.0

1.2. Intended audience

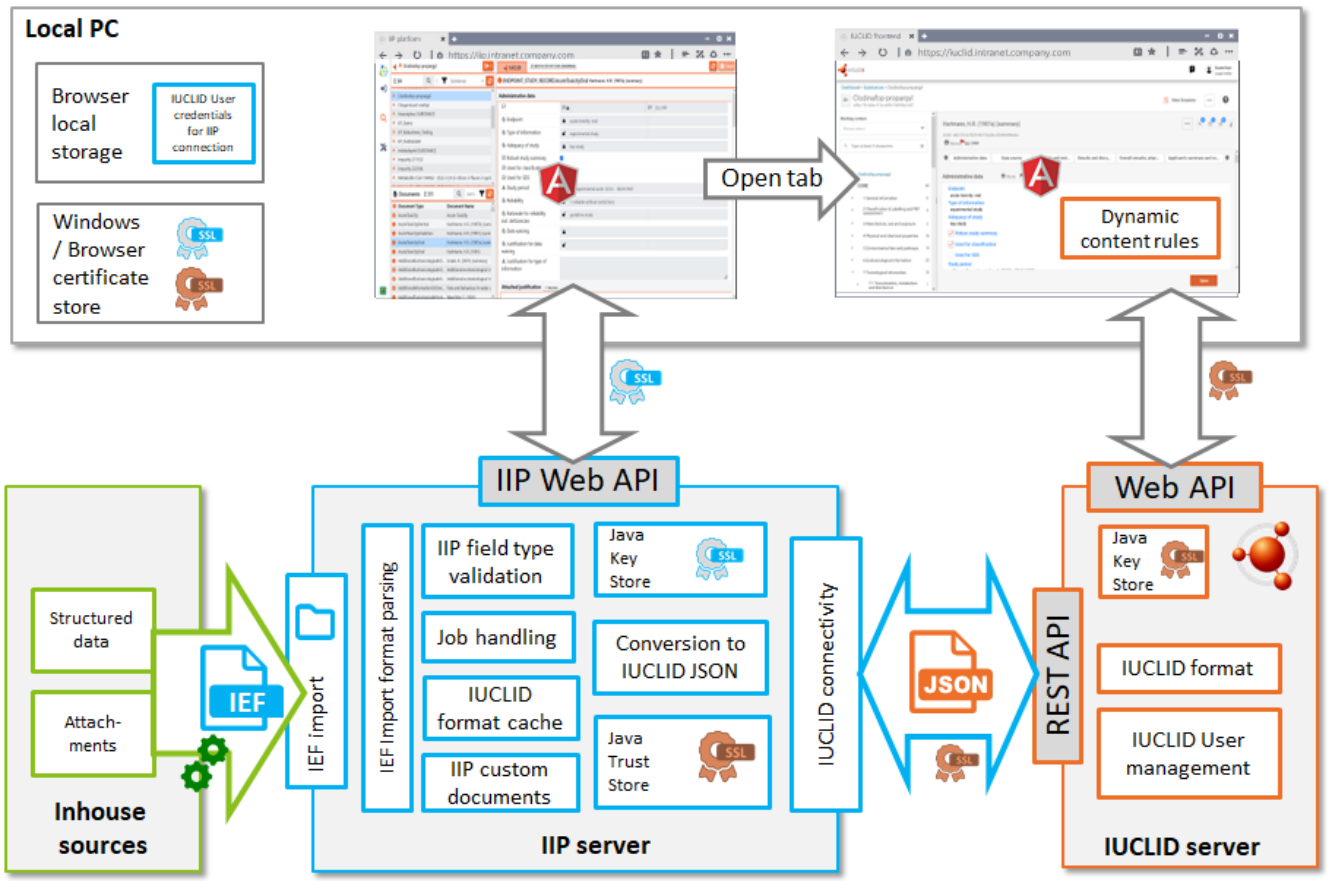
This document is aimed at IT operations staff that need to install or configure the IIP in a particular environment. A good understanding of

- application servers (IIP uses Wildfly)
- concepts like SSL, keystore, truststore and certificates

is helpful.

Chapter 2. IIP architecture

To better understand the installation and configuration requirements, the following figure provides a high-level overview of the IIP server and its environment in a simple form:



The IIP server is depicted in the middle lower part. It has the following connections to surrounding applications:

- The IIP provides its own web interface. For secure HTTPS communication a certificate needs to be installed in the **Java Keystore** of the IIP server.
- For a specific user session the IIP connects via HTTPS to the REST API of a running IUCLID instance. The connection is established via Java and therefore the certificate used to communicate with the IUCLID instance needs to be trusted in the **Java Truststore**.
- The IIP can be used concurrently and consequently the IIP can have muliple parallel sessions to the same or different IUCLID instances.
- The IIP uses two folders for import and export of IEF files, that need to be accessible to both the IIP and the end user. IEF stands for **IIP Exchange format** and is a format definition used be the IIP to import data to / export data from IUCLID. For more information about the IEF format see the separate format manual
- The user can have the IIP web frontend and the IUCLID web frontend open in different tabs of the web browser. These sessions are isolated; the IIP webfrontend has no direct connection to IUCLID neither via the browser nor a direct connection. The only "integration" is the possibility in IIP to open a new browser tab to display a specific document in the IUCLID web interface.

The core technical stack of the IIP is very similar as IUCLID itself: * a Java application deployed as WAR file * an application server (for IIP: Wildfly, formerly Jboss; for IUCLID: Payara, derived from Glassfish) - a Java application deployed on a Wildfly application server and a web frontend (for IIP and IUCLID: Angular2)

```
Please note, as a difference to IUCLID, that the IIP does not have its own database
with IUCLID data, nor its own user management. There is no logon mechanism for the
IIP. For each connection to the IIP via the local browser a separate session is
established and the URL and credentials of the IUCLID instance to connect to is
provided from the browser local storage. Consequently, there is no confidential data
stored on the IIP server. A user can have multiple sessions to the IIP with parallel
connections to different IUCLID instances.
```

2.1. Difference between Java KeyStore and TrustStore

The IIP is a Java application and uses features of the Java plattform. For secure communication, the Java Keystore and TrustStore are relevant. The Keystore and TrustStore are opposite to each other and it is important to understand their differences:

- The TrustStore is used to store certificates from Certified Authorities (CA) that verify the certificate presented by the server in an SSL connection. Hence, it securely identifies other servers - in the IIP case it needs to securely identify the IUCLID instance.
- The Keystore is used to store private key and identity certificates of the IIP server that it should present to a client (here the IIP web interface) to securely identify itself.

See the section "Application Server configuration" for more details.

2.2. Proxy between IIP server and IUCLID instance

Eventually, there is a proxy server between the IIP server and the IUCLID instance(s), that needs to be configured in the application server in order to enable a connection between the IIP and the IUCLID instance(s).

See the section "Application Server configuration" for more details.

Chapter 3. Installation options

The IIP can be installed either on a server or a local PC (desktop, laptop). A server installation is recommended. Please note that the term "IIP server" is used in this manual and may refer both to a server installation or a local installation, as technically it is an installation based on an application server.

The relevance of sections for either installation option is marked in each chapter. If you feel that changes that the installation and configuration is beyond your limits, please please ask your local IT support and hand over this document to them.

Chapter 4. Installation

This chapter is relevant for both server and local installation options

4.1. HW and SW installation requirements

The IIP has been developed and tested with the following configuration:

- Java version, at least version 8 (tested with 32- and 64 bit environments)
- Windows 64bit environment - but the IIP should work also on Linux
- A current web browser version (Chrome, Edge, FireFox)
- Sufficient CPU and RAM (16GB or more, at least 768MB of consecutive space)

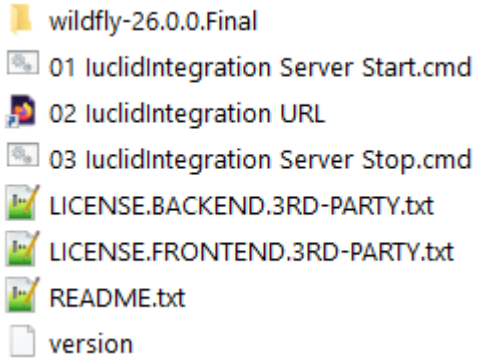
4.2. Supported IUCLID instances

The IIP requires a live connection to a running IUCLID instance and communicates via the REST API (Application Programming Interface) provided by the IUCLID server. Unfortunately, this API is not supported by ECHA for IUCLID instances in the ECHA cloud, due to non-technical reasons. Therefore, the IIP can connect to on-premises / inhouse installation of IUCLID.

4.3. Download and extract

To start, download the zip file from the download directory. The ZIP files contains the IIP application bundled with the Wildfly application server.

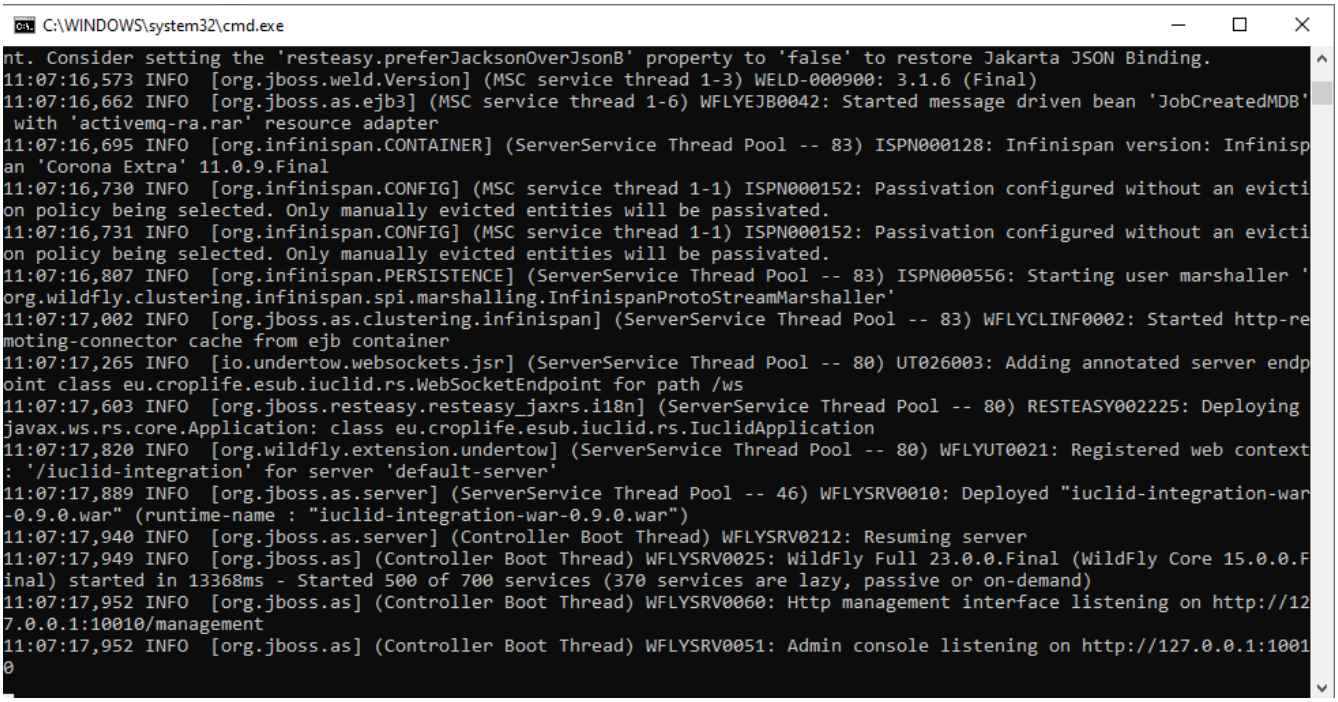
Move the file to a target folder and extract the file. The listing of the root directory looks like this



The name of wildfly folder may change in later releases and is referred in this manual as "wildfly-<*version*>" folder.

4.4. Deployment, first start and initial test

To deploy the application you can click on the start cmd file. A console window opens. The first deployment takes some time and then should similar to the screenshot below.



Please note, that for proper functioning, you must not type into the console window. For productive use we recommend to run the IIP as Windows service (see separate chapter)

Then, to test sucessful deployment, launch the IIP web interface on HTTP:

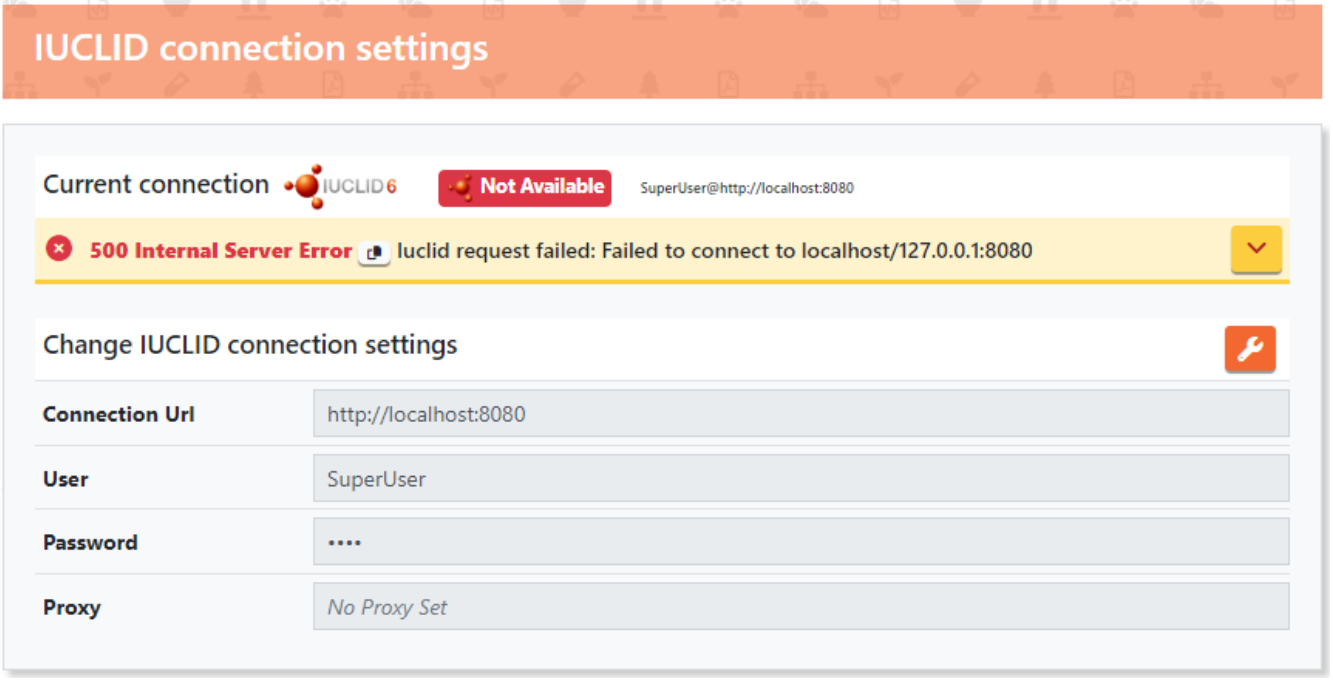
http://localhost:8100/iuclid-integration/

or click on the URL shortcut provided in the directory. The IIP application is listening on port 8100. The IIP web interface should appear.

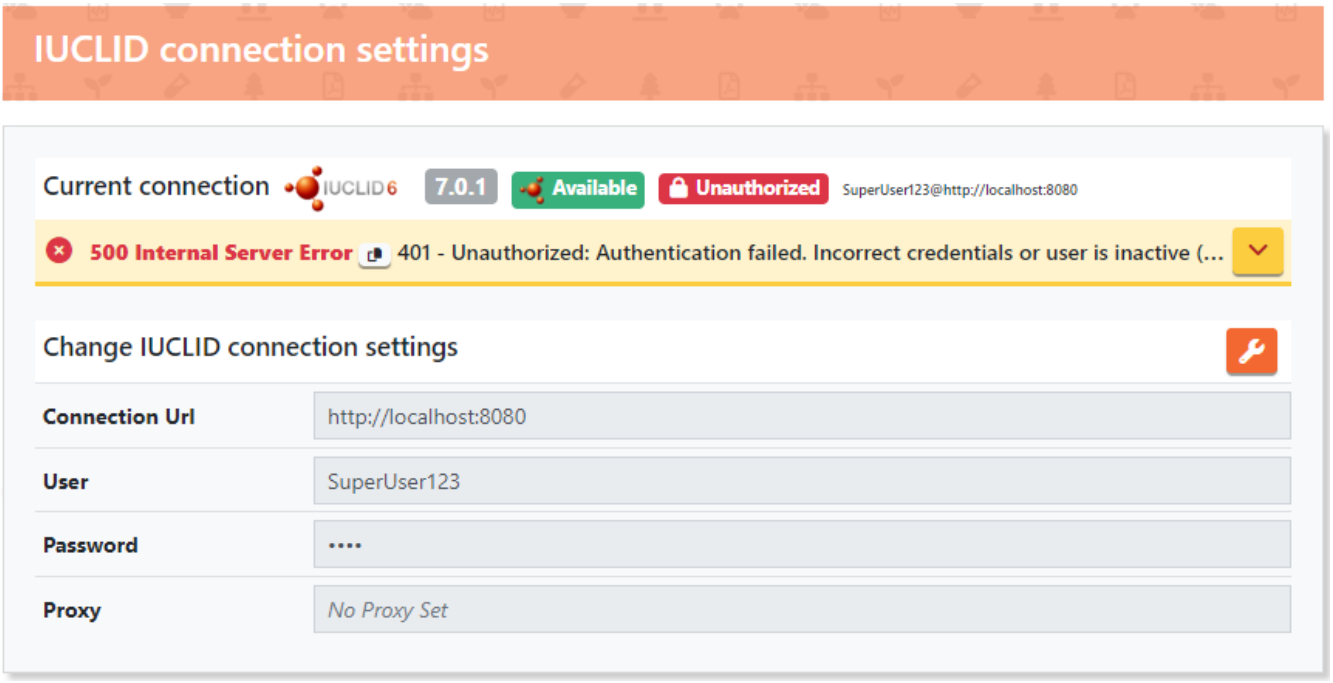
4.5. Test IIP connectivity to IUCLID

In the web interface of the IIP, go to the "Administration" perspective and enter the URL to a IUCLID instance (including "https://" as prefix) and username / password of an existing user in that instance. Please note that the rights to see and modify IUCLID data will via IIP depend 1:1 on the rights as defined in the IUCLID user management.

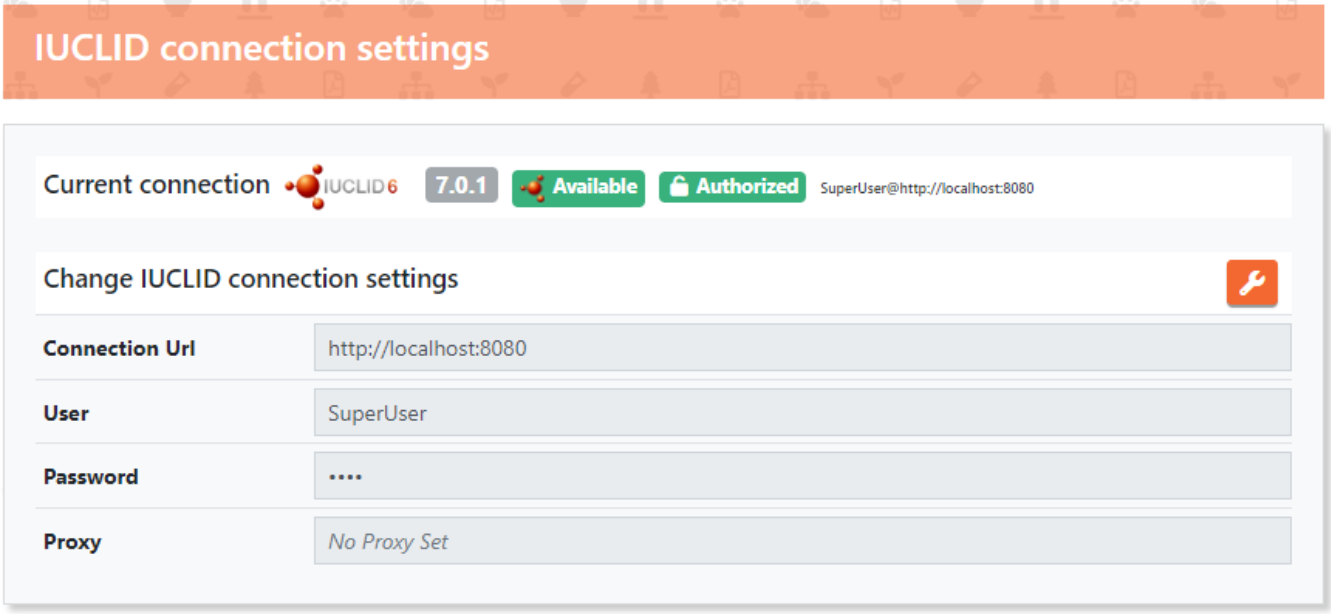
When the current IIP session is not connected to an IUCLID instance the connection symbol is red with status "Not Available":



It turns green with status "Available", when a connection to the IUCLID instance could be established. If the given user is not authorized to access data (tested by accessing the users profile), a permission status "Unauthorized" is shown in red:



If the given user is authorized to access data, the permission status turns green with status "Authorized":



The IIP functionality may only be used with an IUCLID connection that is "Available" and "Authorized".

Please note that you may face issues when connecting to the IUCLID instance via the (recommended!) HTTPS protocol; the following error message may appear:

```
Iuclid request failed: sun.security.validator.ValidatorException: PKIX path building
failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find
valid certification path to requested target
```

In this case you need to add the public SSL certificate of the IUCLID instance to the Java trust store on the IIP server, see chapter Configuration of Java Truststore for more details.

4.6. Using HTTPS to connect to the IIP server

This chapter is informative for local installations. Connections to an IIP server listening on "localhost" do not require a HTTPS connection, an HTTP connection is sufficient.

The IIP port for HTTPS is 8463, but to use HTTPS, an appropriate certificate needs to be added to the Java key store. In productive use, it is recommended to deactive the HTTP protocol for the IIP application on server installations (not mandatory for local installations). Please see the chapter Configuration of the Java Keystore to support SSL connection to web browser for details.

Chapter 5. IIP configuration

This chapter lists configuration tasks, that are specific to the IIP. For other generic tasks related to application servers see the section "Application Server configuration"

Most IIP-specific configurations are to be made in the section system-properties of the Wildfly file “standalone.xml” which can be found in “wildfly-<*version*>\standalone\configuration\standalone.xml” below the root installation directory.

5.1. The file standalone.xml

Below you see an example for the section system-properties in the "standalone.xml" for the IIP version 1.4.0

```
    <system-properties>
        <property name="import.dir" value="${jboss.server.data.dir}\import"/>
        <property name="import.help" value="To select IEF files for IUCLID import by
the IIP, please first copy your IEF data plus linked attachments into the IIP import
folder (mounted as directory '${import.dir}\%s' in the IIP). Then, they can be
selected for processing in this dialog. For questions on how this folder is accessible
and named from your local PC please contact your local IT admininstrator."/>
        <property name="export.dir" value="${jboss.server.data.dir}\export"/>
        <property name="export.help" value="The IEF files with attachments are
exported into the IIP export folder (mounted as directory '${export.dir}\%s' in the
IIP). For questions on how this folder is accessible and named from your local PC
please contact your local IT admininstrator."/>
        <property name="iuclid.demoMode" value="${iuclid.demoMode:false}"/>
        <property name="iuclid.definition.cache"
value="${jboss.server.config.dir}\definitions"/>
        <property name="iuclid.definition.documentation" value="false"/>
        <property name="iuclid.definition.report" value="false"/>
        <property name="version.url"
value="https://esubmission.croplifeeurope.eu/version/"/>
        <property name="help.url"
value="https://esubmission.croplifeeurope.eu/static/iip/UserHelp.html"/>
        <property name="security.trustall" value="false"/>
        <property name="session.timeout" value="30"/>
        <property name="tls.keystore.path" value="iip.keystore"/>
        <property name="tls.keystore.password" value="iip"/>
        <property name="tls.key.password" value="changeit"/>
        <property name="tls.generate.host" value="localhost"/>
        <property name="http.timeout.connect" value="10000"/>
        <property name="http.timeout.read" value="60000"/>
    </system-properties>
```

The individual settings are explained below.

5.2. IEF import directory

For the IEF import functionality, the respective IEF files have to be put below import directory that is both accessible for the user and the IIP. The user first puts IEF files (plus eventually referenced attachments) to be processed below this directory by normal file system operations (e.g. in the Windows Explorer), which subsequently can be picked up via the IIP browser interface in the import operation. Further subfolders can be created as needed. The IIP cannot upload files to this directory or for direct processing via the web browser.

For a server installation this directory has to have the following characteristics and has to be set up by IT admins:

- accessible from the local PCs for all intended users (e.g. via the Windows Explorer)
- directory considered local for the IIP server, either as local directory or drive letter (windows) / mounted drive (Linux)
- access rights to create, modify and delete files for all intended users

For the IIP this directory needs to be configured in “standalone.xml”. The default setting for this “import.dir” is the directory “import” below the default data directory “wildfly-<*version*>\standalone\data”:

```
        <property name="import.dir" value="${jboss.server.data.dir}\import"/>

        <property name="import.help" value="To select IEF files for IUCLID import by
the IIP, please first copy your IEF data plus linked attachments into the IIP import
folder (mounted as directory '${import.dir}\%s' in the IIP). Then, they can be
selected for processing in this dialog. For questions on how this folder is accessible
and named from your local PC please contact your local IT admininstrator."/>
```

This property can be changed to any other directory accessible for the machine (server or local PC) where the IIP is installed. If the shared directory is a network directory, it needs to be accessible locally to the IIP server, e.g. for Windows by mapping it to a drive letter; or for Linux mounting it. During testing on Windows mounted drives with letters (e.g. “Z:\import”) have worked, but not UNC paths (e.g. \\DataServer\import).

All “*.ief” files in the directory structure are then selectable for import by users via the IIP browser interface.

The administration can use a specific property “import.help” in the standalone.xml to have a specific help text displayed in the import mask of the IIP, to support the user to find this directory. This help text should ideally name the import directory as it is accessible for all users from their local PCs.

5.3. IEF export directory

In the very same way, an export folder can be configured in the standalone.xml, where the IIP creates all exports that have attachments. (Exports without attachments are created in the

Downloads folder of the browser.)

```
    <property name="export.dir" value="c:\temp\export"/>

    <property name="export.help" value="The IEF files with attachments are exported
into the IIP export folder (mounted as directory '${export.dir}\%s' in the IIP). For
questions on how this folder is accessible and named from your local PC please contact
your local IT admininstrator."/>
```

5.4. IEF import Demo mode

The IIP supports a demo mode for IEF upload. This means that a specific annotation is used to “tag” instances created by the IIP. The same single annotation is linked to all respective IUCLID entities and has the following field values:

```
Name: Created by IUCLID Integration Platform - DEMO MODE - please do not delete
Authority: IUCLID Integration Platform
```

The demo mode allows to

- quickly see the instances created / updated by the IIP (eventually use the Advanced Search in the IUCLID web interface)
- delete all instances created in demo mode to repeatedly test the functionality of the IIP (For deletion see the panel in the IIP administration perspective, when the demo mode is turned on).

Please note that the deletion will only affect the instances with an annotation with the respective fixed string, not any other instances.

The demo mode can be turned on / off using the property “iuclid.demoMode” with values true/false:

```
    <system-properties>
        <property name="iuclid.demoMode" value="${iuclid.demoMode:false}"/>
        ...
    </system-properties>
```

The default is false - meaning that the Demo Mode is off. Prior to changing the mode, please shutdown the IIP and restart after the change.

5.5. IUCLID format definition cache

The IUCLID document and phrasegroup definitions are costly to retrieve. The IIP caches the definitions as JSON files in the path configured by the system property "iuclid.definition.cache". By default the definition cache is written in the folder "definitions" within the Wildfly standalone/configuration directory.

```
    <system-properties>
        <property name="iuclid.definition.cache"
value="${jboss.server.config.dir}\definitions"/>
        ...
    </system-properties>
```

For every IUCLID version (major and minor) a subdirectory is created, with one JSON for document definitions and one JSON for phrasegroup definitions with a MD5 hash file for each. The IIP distribution packages the definitions of the latest IUCLID versions. On server startup the definition cache directory is read, hashes are validated and the JSON files are read into memory.

As soon as a client tries to connect to an IUCLID instance with a version that is unknown in the cache, the document and phrasegroup definitions are retrieved and written in the cache directory. This occurs once for each new format.

Client actions are blocked, until the definitions for the unknown IUCLID version are properly cached.

As of now, existing definition caches are not deleted. If all connected IUCLID instances are upgraded to later versions, definition caches for outdated IUCLID versions can be deleted.

5.6. IUCLID definition documentation

In addition to the format cache, an XLSX definition & phrasegroup report may be generated. This can be configured with the system property "iuclid.definition.documentation" (disabled by default).

```
    <system-properties>
        <property name="iuclid.definition.documentation" value="false"/>
        ...
    </system-properties>
```

The documentation is generated in a folder "documentation" inside of the cache directory. This documenation is not accessible for end users via the IIP interface, so it must be handed over to users with IT support.

5.7. IUCLID definition report - DO NOT USE IN PRODUCTION

For technical IUCLID format analysis purposes, additional reports als XLSX files can be generated (disabled by default), directly in the cache directory. Those report is usually not of interest to end users.

Please note that the generation of this report temporarily requires a lot of main memory; the standard settings will not be sufficient. it is not advised to apply this setting in productive environments. Change to at least -Xmx4G in the start script to be on the safe side.

```
    <system-properties>
        <property name="iuclid.definition.report" value="false"/>
        ...
    </system-properties>
```

5.8. Version check URL

This URL is used to automatically check when a user connects to the IIP server, whether a new official IIP version is available. If you want to use this feature, please do not change this URL.

```
    <system-properties>
        <property name="version.url"
value="https://esubmission.croplifeeurope.eu/version/"/>
        ...
    </system-properties>
```

5.9. User help/manual

The system property "help.url" references a URL with a user manual. The default value is the official online user help for the IIP and will be extended with each release. If needed, a custom user manual may be configured here. Please contact us, if you need the sources (Format: asciidoc) for the online help for your amended internal help.

```
    <system-properties>
        <property name="help.url"
value="https://esubmission.croplifeeurope.eu/static/iip/UserHelp.html"/>
        ...
    </system-properties>
```

5.10. Session timeout (session.timeout)

The user session that a browser has with the IIP will be cleared after the specified value (in minutes) from the server, e.g. the IIP import jobs (if any) from the session will be cleared.

5.11. http settings

These are settings for timeouts (in milliseconds) for the HTTP(S) connection between the IIP and IUCLID (not between the IIP and the web browser).

- http.timeout.connect (default 10s)
- http.timeout.read (default 30s) In case there are timeouts due to slow bandwidths and/or slow IUCLID response, this value can be increased.

5.12. Further settings for SSL

The following settings affect the usage of SSL between the IIP server and IUCLID.

- tls settings (TLS = Transport Layer Security) for the SSL context, see chapter Change of default SSL context
- security.trustall: If SSL is used for the connection between the IIP and IUCLID: Does the connection trust all presented certificates (true) - or only the ones in the truststore (false). The default value is false.

Chapter 6. Application server configuration

This chapter lists configuration tasks / possibilities on the level of the application server, not specific for the IIP, but required for proper operation.

This section is provided as additional information, as the application server configuration tasks are standard tasks as for any application deployed in Wildfly and **not specific to the IIP**. The CLE eSEG does not provide any further support related to those activities.

The tasks described in this chapter are likely not mandatory for local installations, only marked explicitly.

6.1. Application server administration

The web interface for the application server can be opened on port 4848, e.g. http://localhost:4848/. This interface can be used for various tasks, e.g. changing the port of the IIP application. Alternatively, the application server configuration files (eg.standalone.xml) can be edited directly. Please check the standard documentation for more details.

6.2. Installation as a Windows service

We recommend that you use the scripts that Wildfly provides to install the IIP as a service on Windows.

You will need the following steps:

- Firstly, move to the subfolder “wildfly-<*version*>\docs\contrib\scripts\”.
- Next, copy the “service” directory to wildfly-<*version*>\bin
- Then, open an Admin Shell and navigate into the “service” folder
- Run “service.bat install“

See e.g. http://www.mastertheboss.com/jbossas/jboss-configuration/installing-wildfly-on-windows/ for additional info.

Before running the service.bat you may adjust settings in the script, e.g. the display name of the service.The service will be installed in a way that “wildfly-<*version*>\bin\standalone.bat” is used for the server process execution. You may influence the runtime settings, e.g. max memory, by changing the JAVA_OPTS environment setting in “wildfly-<*version*>\bin\standalone.conf.bat”

6.3. IIP Ports

By default the port 8100 is used for HTTP and port 8463 is used for HTTPS.

To change the default ports, the environment variables “jboss.http.port” and “jboss.https.port” in the standalone.xml can be changed:

```
<socket-binding-group name="standard-sockets" default-interface="public" port-
offset="${jboss.socket.binding.port-offset:0}">
    ...
    <socket-binding name="http" port="${jboss.http.port:8100}"/>
    <socket-binding name="https" port="${jboss.https.port:8463}"/>
    ...
</socket-binding-group>
```

6.4. Proxy server

Eventually relevant for local installations, if you do not manage to get a connection between the IIP and your IUCLID instance.

If the HTTP(S) connection between the IIP server and the IUCLID instances requires a proxy, this needs to be configured in the IIP application server Wildfly. Configuring WildFly to use Proxy Host settings is not different from any other Java application. Basically you need to include the following System Properties in your start script:

```
http.proxyHost: the host name of the proxy server
http.proxyPort: the port number, the default value being 80.
http.nonProxyHosts:a list of hosts that should be reached directly, bypassing the
proxy. This is a list of patterns separated by □|□.
http.proxyUser=The proxy user name if needed
http.proxyPassword=The proxy password if needed.
```

Please refer to the documentation of Wildfly for more details.

6.5. Configuration of the Java Keystore to support SSL connection to web browser

The Java Keystore is used to store private key and identity certificates of the IIP server that it should present to a client (here the IIP web interface) to securely identify itself via SSL.

The IIP server is a Java application, therefore the Windows certificates are not accessible for the IIP and a separate keystore is needed.

To enable SSL communication to the client, the following steps must be done:

- a domain certificate must be obtained or created
- a Java Keystore must be created on the IIP server, if not already present
- the certificate must be added to the Java Keystore
- the default SSL context is switched to signed certificates

The folloing steps are **sample** steps to illustrate the process. Please modify and

> adapt to your local needs and policies!

6.5.1. Creation of a domain certificate

This process is company specific and out of scope of this manual.

6.5.2. Creation of Keystore

- Create a keystore in the directory "wildfly-<*version*>\standalone\configuration". The keystore stores sensitive information and should therefore be created in a safe place with access restrictions.
- The parameters to create the keystore are as following:
  - Keystore parameters
  - Expand source
- Choose **Create a new Keystore** and for **Keystore Type** select **JSK option**
- In window **Generate Key Pair** choose **RSA** with Key Size **2048**.
- In window **Generate Key Pair Certificate** under **Validity** Period enter **1**.
- In window **Name** set parameters for **CN,OU,O,L,ST,C**
- In window **New Key Pair Entry Alias** Enter Alias **localhost**
- In window **New Key Pair Entry Password** enter/confirm password **changeit**

6.5.3. Import of domain certificate

Now the SSL certificates are imported into the newly created keystore. SSL certificates are best imported in *.p7b format (it contains all certificates neccessary).

- Right click on **localhost** and choose **Import CA Reply** > **From File** >, then select the appropriate p7b file and click on Import button.

6.5.4. Change of default SSL context

The SSL configuration is configured in the file standalone.xml. By default, the application server contains the configuration for an unsigned SSL certificate for testing. To change the configuration there are two options: * usage of the Elytron subsystem (not further discussed here) * direct editing of the standalone.xml file

With manual editing, the following sections in standalone.xml must be edited:

- tls
  - tls\key-stores\keystore
  - tls\key-managers\key-manager
  - tls\server-ssl-contexts\server-ssl-context
- https-listener XXXX ??

- interfaces XXXX ???
- system-properties
  - property name="tls.keystore.path"
  - property name="tls.keystore.password"
  - property name="tls.key.password"
  - property name="tls.generate.host"

6.6. Configuration of Java Truststore

This chapter is ony relevant for local installations, if you do not manage to get a connection between the IIP and your IUCLID instance and get an error message similar as below.

Whenever Java attempts to connect to another application over SSL (e.g.: HTTPS, IMAPS, LDAPS), it will only be able to connect to applications it can trust. The way trust is handled in Java is that you have a truststore (typically $JAVA_HOME/lib/security/cacerts) in JKS format. The truststore contains a list of all known Certificate Authority (CA) certificates, and Java will only trust certificates that are signed by one of those CAs or public certificates that exist within that truststore.

In case the IUCLID instance has a public certificate signed by a CA that is not trusted by the truststore, the following error message from Java will appear in the Administration perspective in the IIP web interface, when trying to connect:

```
Iuclid request failed: sun.security.validator.ValidatorException: PKIX path building
failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find
valid certification path to requested target
```

To solve this problem you have to import the public certificate of the target instance into the truststore (usually the file "cacerts" managed by your Java runtime). Please make sure you use the right truststore; you may have eventually multiple Java environments installed. Then please check that the right truststore is used.

The details of those steps are specific to the installed Java environment on the IIP server, please refer to your individual documentation.

Chapter 7. Useful links and tools

- IIP download section - https://esubmission.croplifeeurope.eu/iip/download/
- Latest wildfly documentation - https://docs.jboss.org/author/display/WFLY/Simple%20SSL%20Migration.html
- SSL implementation on wildfly - https://wildfly-security.github.io/wildfly-elytron/blog/auto-self-signed- certificate-generation/
- How to configure SSL/HTTPS on WildFly - http://www.mastertheboss.com/jbossas/jboss-security/complete- tutorial-for-configuring-ssl-https-on-wildfly/
- CSR generate - https://www.namecheap.com/support/knowledgebase/article.aspx/9854/2290/how-to-generate-a-csr-code-on-a-windowsbased-server-without-iis-manager/

Useful tools - use at your own risk!

- Keystore explorer - https://keystore-explorer.org/
- Online (!) SSL converter - https://www.sslshopper.com/ssl-converter.html