

3D Juump Infinite Administration Manual

[AM_EN] version 3.3

Introduction	9
Disclaimer.....	11
1 - Documentation coverage	11
2 - System security is under the Licensee's responsibility	12
3 - Documentation liability	12
4 - Third-party software components.....	12
Changes.....	13
Overview.....	14
1 - Introduction.....	14
2 - Definitions	15
2.1 - Digital mock-up (DMU).....	15
2.2 - 3D Juump Infinite infrastructure	16
3 - Components	18
4 - DMU Flow	19
Roles	23
1 - ∞Directory.....	23
2 - ∞Server/∞Proxy/∞StaticProxy	25
2.1 - ∞Proxy	27
2.2 - ∞Server.....	27
2.3 - ∞StaticProxy.....	27
Access-rights	28
1 - Overview	28
2 - Tags	28
3 - Access Rights	29
4 - Rules.....	29
4.1 - Access Rights Inheritance and Validation.....	29
4.2 - User using an application	30
4.3 - User connection to a proxy	30

4.4 - User access to a build	30
4.5 - User assets administration.....	30
4.6 - User licence borrowing.....	30
4.7 - Third party application user access rights	30
4.8 - Proxy build replication	30
Installation	32
1 - Infinite services installation.....	32
1.1 - Requirements	32
1.2 - Third party service version	33
1.3 - Install scenarios.....	34
1.4 - Test the ∞Directory front-end web page.....	34
2 - LMX (optional, for borrow support).....	34
2.1 - Run a LM-X server (for borrow support)	34
2.1.1 - Install the LMX borrow license file.....	35
2.1.2 - Extract the ∞Directory keys.....	35
3 - OpenId Connect	35
3.1 - Configuration.....	35
3.2 - User identification	36
3.3 - Id token usage	36
3.4 - Authentication webhook	36
3.5 - Access Token usage	36
3.6 - Non standard query parameters	37
3.7 - Example.....	37
Administration.....	38
1 - Web application well known urls	38
2 - Administation web application interface	39
2.1 - Login.....	39
2.2 - Dashboard	40
2.3 - Team Management	40

2.3.1 - Users	40
2.3.2 - Teams.....	42
2.4 - Network.....	44
2.4.1 - Applications	44
2.4.2 - Servers	46
2.4.3 - Connections.....	50
2.5 - Projects.....	53
2.5.1 - Create Projects	54
2.5.2 - Check a Project Status	54
2.5.3 - Request a Build	55
2.5.4 - Synchronize metadata.....	55
2.5.5 - Get descriptor.....	55
2.5.6 - Clean generation data	56
2.5.7 - List Builds.....	56
2.5.8 - Delete a Build	56
2.5.9 - Export a Build	56
2.5.10 - Delete Projects.....	56
2.6 - Settings.....	56
2.7 - Licences.....	57
2.7.1 - Activity report	58
2.7.2 - Install 3D juump infinite licence	59
3 - Monitoring.....	59
3.1 - Status monitoring.....	59
3.2 - Error notification (Webhook)	59
4 - Data access control.....	60
4.1 - Tcp port usage	60
4.2 - List of service startup users.....	61
4.3 - Data encryption	62
4.4 - Data caching policy	62

4.5 - Folder architecture	62
4.6 - ∞Server.....	63
4.6.1 - DMU Data	63
4.6.2 - Administration Data.....	63
4.7 - ∞Proxy and ∞StaticProxy	63
4.7.1 - DMU Data	63
4.7.2 - Administration Data.....	64
4.8 - ∞Directory	64
4.8.1 - Administration data	64
4.8.2 - Assets	64
4.9 - 3D Juump Infinite Client Application.....	64
4.9.1 - Connected mode.....	64
4.9.2 - Standalone mode.....	64
4.10 - Communications.....	64
4.11 - DMU files	65
5 - Fault tolerance and preventive maintenance.....	65
5.1 - Temporary network failure.....	65
5.2 - ∞Proxy or ∞Server failure	65
5.2.1 - Redundancy by ∞Proxy	65
5.2.2 - Build backups	65
5.2.3 - Project backups.....	65
5.2.4 - ∞Server backups	65
5.2.5 - Asset databases	65
5.3 - ∞Directory failure	66
5.3.1 - Administration web interface	66
5.3.2 - Authentication service	66
5.4 - LM-X failure	67
6 - Migration procedures.....	67
6.1 - Cluster migration.....	67

6.2 - Assets migration	67
Native client.....	69
1 - Modes	69
1.1.1 - Online mode	69
1.1.2 - Standalone mode	70
2 - Host requirement	70
2.1 - Hardware	70
2.2 - Software	71
3 - Installation	71
3.1 - Installing the <i>Hub</i> and <i>Browser</i> add-on	71
3.1.1 - Requirements	71
3.1.2 - Installation procedure	71
3.1.3 - Default settings configuration	72
3.2 - Installing the <i>Local DMU Manager</i> add-on.....	73
3.2.1 - Requirements	73
3.2.2 - Installation procedure	73
3.2.3 - Local web application	73
3.2.4 - ElasticSearch configuration	73
Third party application protection.....	74
1 - Defining user access rights	74
2 - Request scope	74
3 - Protect your web application	75
4 - Validate token and claims	75
Annexes	77
1 - Release notes	77
1.1 - 02/16/2025: version 3.3.16	77
1.2 - 18/02/2024: version 3.3.15	78
1.3 - 26/01/2024: version 3.3.14	78
1.4 - 16/01/2024: version 3.3.13	78

1.5 - 14/12/2023: version 3.3.12	78
1.6 - 8/12/2023: version 3.3.11	78
1.7 - 10/10/2022: version 3.3.10	79
1.8 - 23/05/2022: version 3.3.9	79
1.9 - 09/05/2022: version 3.3.8	79
1.10 - 02/05/2022: version 3.3.7.....	79
1.11 - 01/04/2022: version 3.3.6.....	80
1.12 - 15/03/2022: version 3.3.5.....	80
1.13 - 10/03/2022: version 3.3.4.....	80
1.14 - 26/02/2022: version 3.3.3.....	81
1.15 - 26/01/2022: version 3.3.2.....	81
1.16 - 03/01/2022: version 3.3.1.....	82
1.17 - 01/12/2021: version 3.3.0.....	82
2 - Security guidelines	83
3 - Third-party software licenses.....	83
4 - Range of use	83
4.1 - Minimum requirements.....	83
4.2 - Supported input formats	84
4.2.1 - Geometry.....	85
4.2.2 - Metadata.....	85
4.2.3 - Annotation.....	85
4.3 - Supported output formats	85
4.3.1 - Geometry.....	85
4.3.2 - Image	86
4.3.3 - Metadata.....	86
4.3.4 - Presentation	86
4.4 - Limits	86
5 - Export control classification.....	87

Introduction

This documentation is targeted to your system administrators and CAD database specialists.

- For the *system administrator*, this documentation describes the installation and administration of 3D Juump Infinite back-end servers and their underlying third party components. The main features of the administration web interface manual are also presented to cover most of the administration use cases: create users, groups and configure the generation of the data sets, setup replication.
- For the *CAD database specialist*, the data set generation procedure is detailed to let you build your own data from your information system back-ends.

This documentation covers the installation of the 3D Juump Infinite server components:

- ∞Directory, connecting all the components of the 3D Juump Infinite; with the web administration interface;
- ∞Server, in charge of the compilation, optimization and packaging of the data sets;
- ∞Proxy, a proximity relay for data set broadcasting;
- ∞StaticProxy, a stub server for broadcasting data set from packaged files;

An Administration chapter is provided to explain the security policy and describe some maintenance points.

The last part is dedicated to 3D Juump client application deployment:

- 3D Juump Infinite client application installation
- Installer customization

For a primer installation, we suggest to start with the setup procedure of an ∞Directory and an ∞Server installations, validate it with the web administration interface. Then, if you want to construct your own data set, read the chapter dedicated to the data set generation procedure. Eventually, read the last chapter to deploy the client applications on your end-user host machines. The authentication of the users on the 3djuump infinite system is forwarded to an OpenID Connect authentication server (Authenticator). The ∞Directory does not hold any authentication mechanism, but defers such authentication. There are public servers that can provide such an authentication mechanism, for example auth0 (<https://auth0.com/>). The system administrator may choose an opensource solution hosted on its own server infrastructure such as keycloak (<https://www.keycloak.org/>).

Disclaimer

1 - Documentation coverage

The information outlined in this documentation is intended to be used for the following purpose:

- installation of 3D Juump Infinite back-end and front-end;
- creation of a specialized 3D Juump Infinite data sets specific to customer needs;
- diffusion of 3D Juump Infinite data sets by 3D Juump Infinite back-end;
- administration of 3D Juump Infinite users, groups and components.

The Licensee is liable for any damage or injury to any person or entity arising out by or in connection with its developed works (piece of software developed for/with 3D Juump Infinite) and its data sets (generated for/by 3D Juump Infinite).

The developed works and data sets shall be conformed to design and implementation guidelines and restrictions described in the Documentation.

The software functions available for development are documented. The Licensee shall not utilize the undocumented functionalities.

The Licensee shall be compliant with the 3D Juump Infinite administration, integration and operation recommendations stipulated in the Documentation. AKKODIS INGENIERIE PRODUIT

SAS shall not be held responsible in case of any damage caused by the non-compliance to these recommendations.

2 - System security is under the Licensee's responsibility

The Licensee's system has to be compliant with the state of the art and the Documentation's requirements, especially in terms of security. It is reminded to the Licensee that it has to anticipate safety plans and measures to minimize the consequences notably linked to a possible temporary disruption or a data loss generated by the Software or by any security breach in the Licensee infrastructure which hosts 3D Juump Infinite.

The Licensee shall be held responsible in case of the developed works and data sets compromise the integrity, performance and security of the 3D Juump Infinite.

3 - Documentation liability

The source code and template examples provided through this Documentation are only intended to illustrate. They shall not be used by the Licensee, as developed works or data sets. If they are, AKKODIS INGENIERIE PRODUIT SAS declines any responsibility and any warranty is excluded.

Although reasonable effort is made to ensure that the information in the Documentation is complete and accurate at the time of release, AKKODIS INGENIERIE PRODUIT SAS cannot assume responsibility for any existing errors. Changes and/or corrections may be incorporated in future versions.

4 - Third-party software components

The following software components are provided along with 3D Juump Infinite installers and scripts:

- Executables and plugins located in third-party subfolder,
- Libraries and their dependencies (enumerated in the licensing document).

The exhaustive list of third-party components and the author's identity is made available by AKKODIS INGENIERIE PRODUIT SAS in the menu of the Software, under the link "About". The Licensee is responsible for their proper installation, configuration and usage in compliance with the license of each third-party software component and with its proper software deployment and security policies.

AKKODIS INGENIERIE PRODUIT SAS shall not assume responsibility for bug or operating disruption caused by these third-party software components. Updated versions might be incorporated in future versions.

Changes

This annex summarizes the changes in the documentation since the previous version

- LMX is now optional. It is required only to get borrow support with the local DMU Manager.
- Supported LTS distribution of Linux are now Debian 11 (bullseye) and Ubuntu 20.04 (focal).
- Assets are now saved in the ∞Directory postgres database. It is up to the integrator to add redundancy and backup.
- The OpenIdConnect configuration has changed.

Overview

1 - Introduction

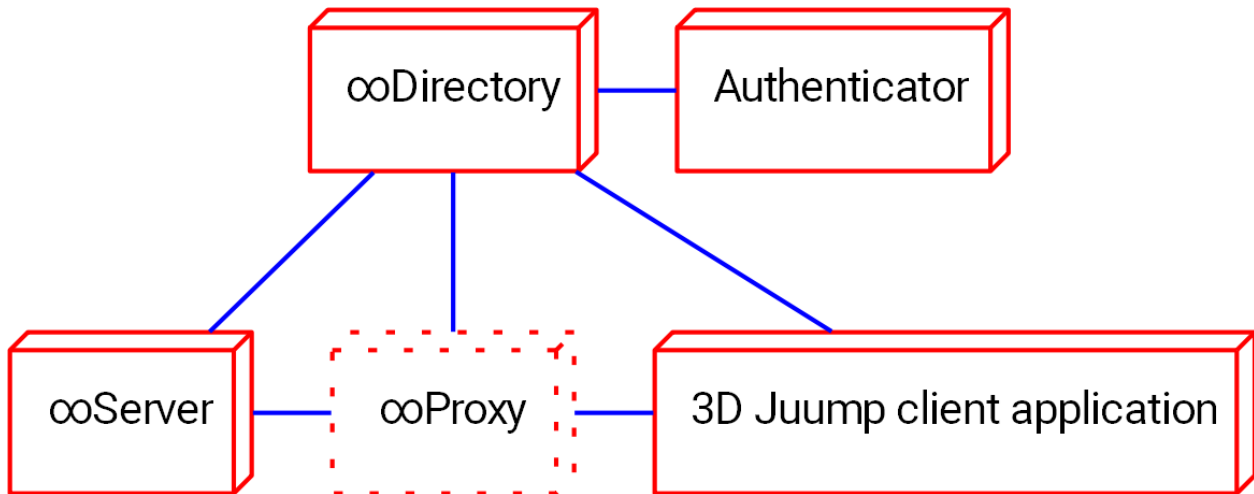
3D Juump Infinite is a software suite allowing the users to browse an entire DMU¹ in 3D on a standard PC².

It relies on a publication server (or **∞Server**), responsible for the preparation and optimization of the CAD source data, a network of relays (or **∞Proxy**) for broadcasting, and a central management server (or **∞Directory**).

On the end user side, the 3D Juump Infinite DMU client application browser is simply called **3D Juump Infinite**.

¹ Digital Mock-Up

² Personal Computer



3D Juump servers and client

2 - Definitions

In order to explain the design of 3D Juump Infinite, it is mandatory to introduce several keywords related to the DMU and CAD product.

2.1 - Digital mock-up (DMU)

The digital mock-up (DMU) is composed of multiple elements:

Digital mock-up

A mock-up is a partial or complete representation of a system in order to preview, test or validate its aspects or its behavior. A digital mock-up is built from computer files storing a tree of three dimension geometries, displayed with a 3D rendering software, in our case, 3D Juump.

Part

A model or template, be it an assembly or a single element. This part is referenced by the digital mock-up but is not present (*instanced*) within it. It is not localized in the digital mock-up. Ex: a wheel.

Part Instance

One instance of a *part*, be it an assembly or a single element. This instance has a 3D representation located in the digital mock-up. Ex: the front-right wheel.

Product Structure

A hierarchy of *part instances*. Ex: a car with four wheels.

Part Number

The unique identifier of a *part*.

Metadata

Any textual or numeric information decorating a *part* or *part instance*, usually presented as a *key=value* pair. Ex: *provider='WheelEx Inc.'*.

Annotation

Any textual or graphical information decorating a *part* or *part instance*, represented in 3D as floating labels. Ex: contextual directions for assembly, tolerancing, etc.

Effectivity

A global parameter or option that changes the way the DMU should be assembled. Ex: What is the car drive-train? Is this car a 4-wheel or a 2-wheel drive vehicle?

Configuration

A set of effectivities. A digital mock-up in a given configuration is also called a configured digital mock-up. Ex: a 4-wheel, metallic paint, 150 hp engine car.

2.2 - 3D Juump Infinite infrastructure

Other keywords are specific to the 3D Juump Infinite infrastructure:

Project

An incremental set of data describing a DMU. It usually corresponds to a product (or product-line) level assembly. For instance, a vehicle manufacturer would probably opt for one project per car model.

Build

An optimized packaged DMU snapshot ready for publication. The DMU processed by 3D Juump Infinite back-end components and served to 3D Juump Infinite front-end client applications.

Connector

A piece of software that feeds a 3D Juump Infinite *project* with data describing the DMU.

DMU data source provider

Any component of your *information system* able to publish DMU data encompassing the part geometry, metadata or effectivity. These components could be a conjunction of your

CMS³, CRM⁴, your ERP⁵, your PDM⁶, your PLM⁷ or simply a file system folder containing all your geometry part files extracted from your CAD⁸ software.

Document

A document is a JSON⁹ exchange file used by the *Connector* and the 3D Juump Infinite back-end to transmit and store data bound to DMU *build* generation. JSON is an open standard human-readable text used to transmit data objects consisting of attribute-value pairs. Thus, this format is also used by 3D Juump Infinite third-party software and the 3D Juump Infinite front-end to exchange data.

DMU flow

A chain of your enterprise components and 3D Juump Infinite back-end components, in charge of DMU extraction from *DMU data source providers*, processing & broadcasting of DMU *builds*.

User

A person that uses 3D Juump Infinite, either through its web administration interface, through a 3D Juump Infinite client application or through one of the offered API. All users must be properly authenticated before being authorized to use 3D Juump Infinite.

Authenticator

A third-party server in charge with the user authentication. 3D Juump Infinite relies on OpenID Connect identity layer to delegate authentication.

Administrator

A user which can logon on the ∞Directory web administration interface. He is able to configure the *DMU flows*, to trigger the generation of DMU *builds* and create *users* and *teams*.

Tag

A keyword used to decorate a *build*, a *user* or a server/proxy component and which defines the access rights to the DMU.

³ Content Management System

⁴ Customer Relationship Management

⁵ Enterprise Resource Planning

⁶ Product Data Management

⁷ Product Lifecycle Management

⁸ Computer Aided Design

⁹ JavaScript Object Notation

Team

A set of users. It mainly acts as a helper concept that applies a common list of tags to its users.

Asset

Any 3D Juump Infinite client application setting (bookmarks, visibility layers, export configurations...) which can be created, manipulated and shared amongst 3D Juump Infinite *users* thanks to 3D Juump Infinite back-end components.

3 - Components

3D Juump Infinite is composed of several software entities.

∞Server

An ∞Server is in charge of the optimization and the packaging of the DMUs (or *projects*) into distributable *builds*. To let *connectors* declare DMUs, the ∞Server publishes a *connector API*.

∞Proxy

An ∞Proxy acts as a proximity relay for DMU broadcasting. It replicates *builds* from ∞Servers or other ∞Proxies. Since an ∞Server is itself an ∞Proxy, the deployment of an ∞Proxy is *not necessary* if your information system and network do not require such DMU broadcasting.

∞Directory

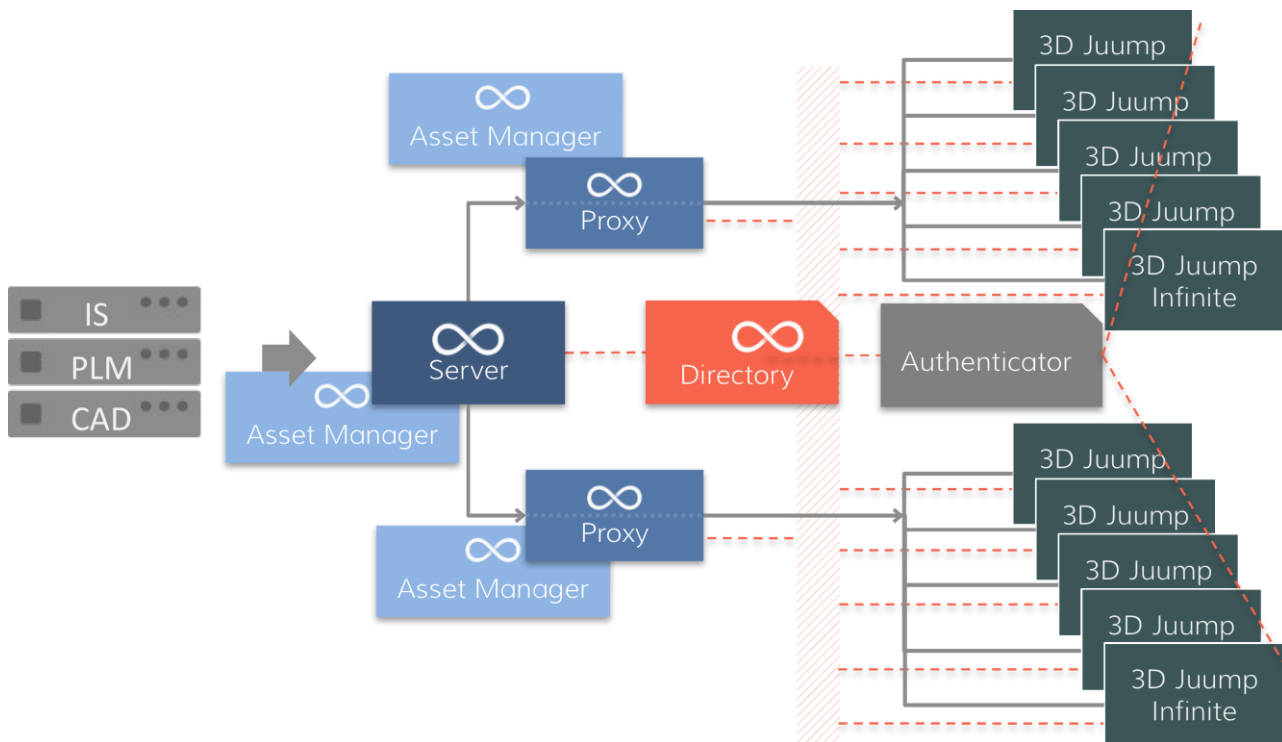
The ∞Directory is in charge of the security management. It monitors and defines the whole 3D Juump Infinite network and is responsible for access-rights management. This management is accessible via a *web administration interface* and programmable via the *directory API*.

3D Juump Infinite client application

The 3D Juump Infinite client application is the DMU browser itself, called *3D Juump Infinite*, when this name is unambiguous.

Web client or Native client

The 3D Juump Infinite client application comes in two flavors. One is the legacy native application (called "Native client") and the other is the new web-based application (called "Web client").



Overview

The previously presented software relies on several third-party components including databases and servers. In particular:

- a [PostgreSQL](#) service,
- an [ElasticSearch](#) service,
- an [Apache](#) HTTP service,
- an optional [LM-X](#) service.

PostgreSQL (or "Postgres") is an SQL object-relational database management system (ORDBMS). It is used by 3D Juump Infinite ∞Directory, ∞Server and ∞Proxy as underlying database engine.

Elasticsearch is a distributed, multitenant-capable full-text search server with a RESTful web interface and schema-free JSON documents. Elasticsearch is built on top of Apache Lucene, developed in Java and is released as open source under the terms of the Apache License. It provides full-text search capabilities to the 3D Juump Infinite client application.

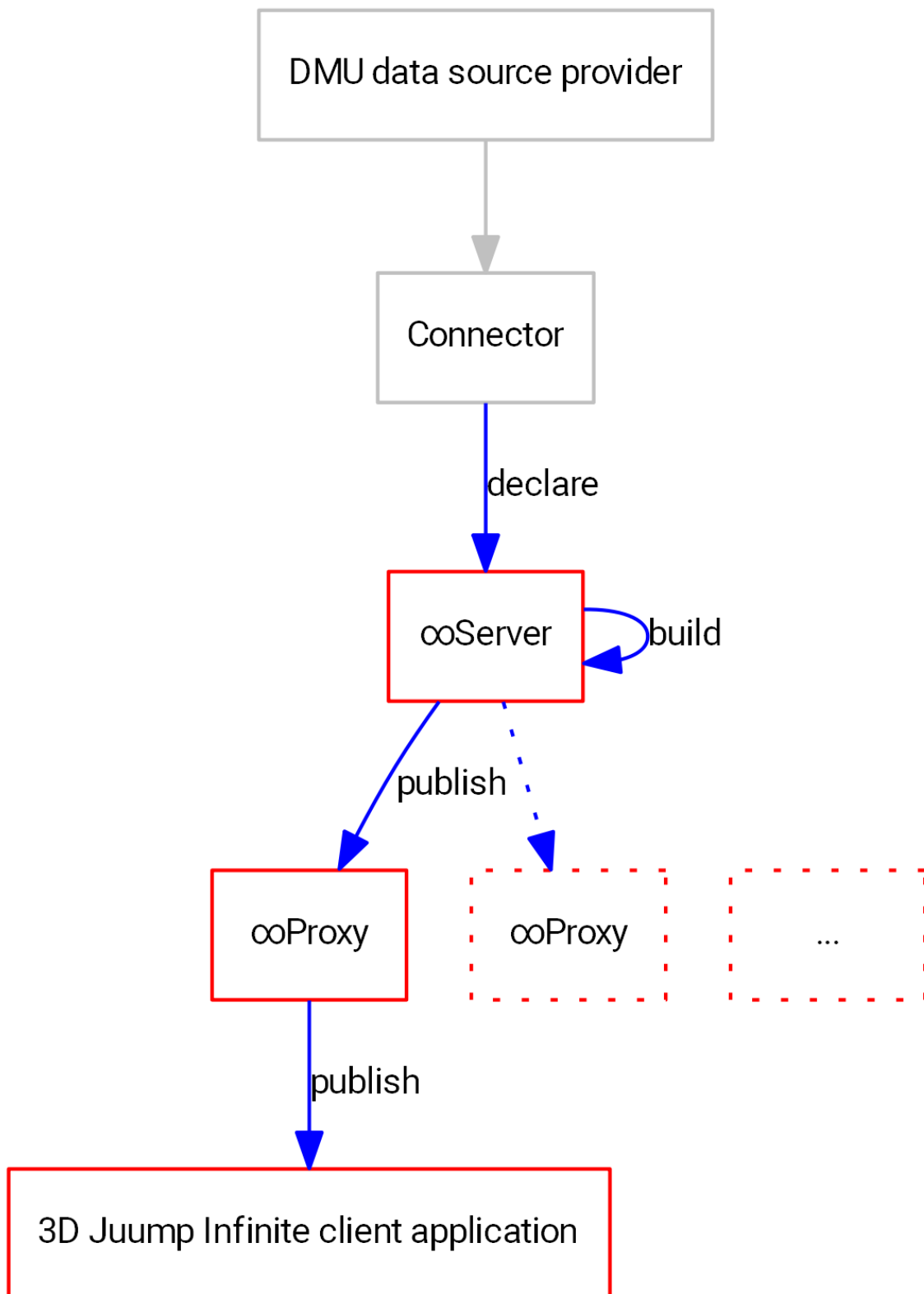
LM-X is a software licensing solution.

4 - DMU Flow

From the enterprise CAD data source to the 3D Juump Infinite client application, the DMU follows several steps:

- First, your enterprise defined **Connector**, bound to your DMU data source providers, incrementally declares the product structure and the associated metadata, effectivities and configurations to a *project* hosted on an ∞Server.

- Once the declared product structure is coherent, the connector triggers a *build*.
- The **∞Server** compiles the product structure and the associated data into a hierarchical database where every leaf points to a geometry.
- Then the ∞Server incrementally pulls the geometry files from the connector and builds an optimized package.
- This *build* is then published on the **∞Proxies** associated to this ∞Server and every connected ∞Proxy then starts to replicate the build locally, in order to obtain a cascade publication.



DMU flow

The Connector must be specified and realized upon your requirements. 3D Juump Infinite only provides interfaces and helpers to feed the ∞Server with such data. In the figure, the enterprise DMU data source providers and your Connector are in gray.

During this process, the ∞**Directory** keeps in touch with every server, be it the ∞Server or the network of relay ∞Proxies. Thus, it is able to handle 3D Juump Infinite client applications connection requests by routing them to the proper ∞Proxy.

For the sake of simplicity, we have described *one* DMU flow. 3D Juump Infinite let you define *several* DMU flows built upon your combination of Connectors bound to your DMU data source providers, attached to several ∞Servers and ∞Proxies.

Roles

1 - ∞Directory

The ∞Directory is in charge of the security management. It monitors the whole 3D Juump Infinite network, from the ∞Servers to the relay ∞Proxies and up to the 3D Juump Infinite client applications. None of these software entities is able to work without the explicit authorization of the ∞Directory. It is also responsible for access-rights management, though it delegates user authentication to a third-party identity service.

The ∞Directory relies on several software components:

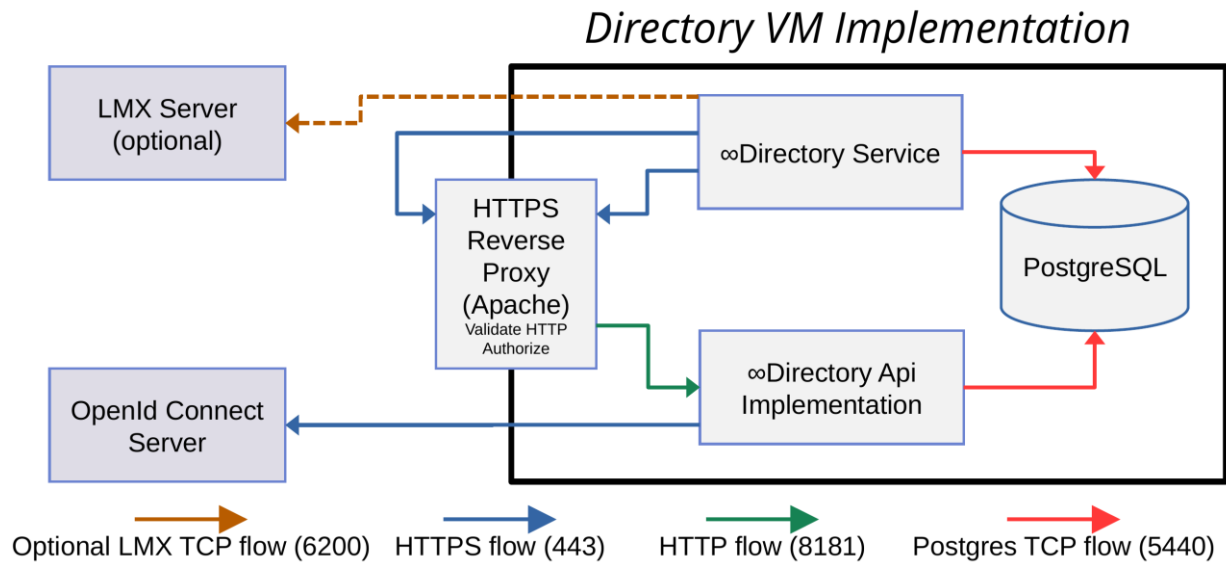
- an optional LM-X server,
- a PostgreSQL server,
- an Apache HTTP server,
- a dedicated service with the provided installers.

The ∞Directory is operated thanks to a web-application (or *front-end*) built upon a web-based API (or *Directory API*) hosted on the HTTP server. The use of this web interface is described in the administration chapter. The API is described in a dedicated document.

The ∞Directory delegates user authentication through OpenID Connect identity layer. Only authenticated users with proper access rights can operate the *Directory API* and the *front-end*, or open a DMU through one of the 3D Juump Infinite client applications and API.

Databases located on the ∞Directory are not directly visible to client-side users. Instead, several API are available and only accessible to authenticated users with proper access rights. Cluster-side server-to-server communication relies on HTTPS.

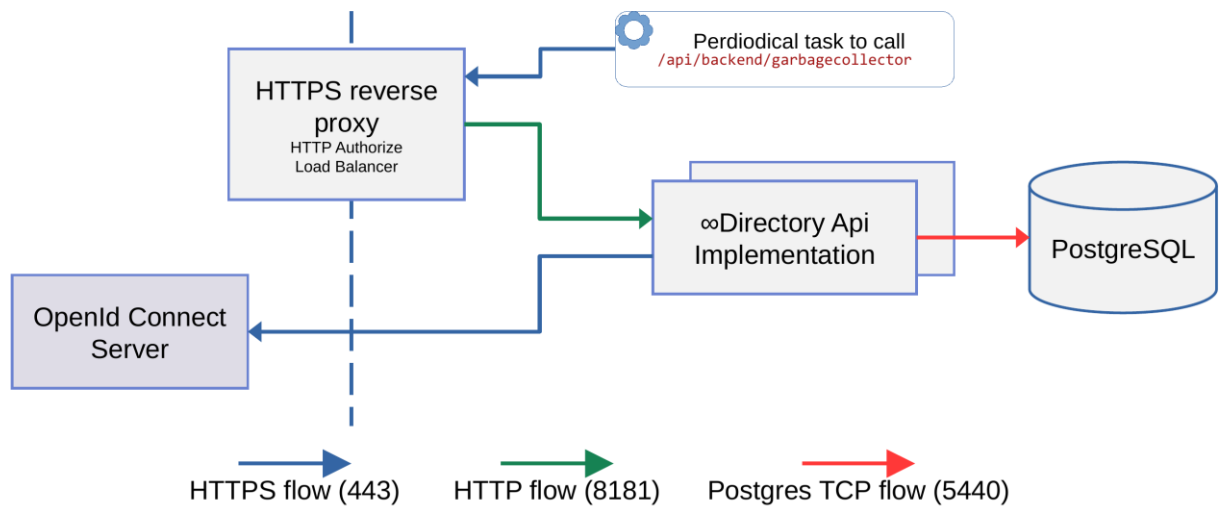
By default the ∞Directory is implemented as a single virtual machine installed using provided scripts.



∞Directory

Alternatively the ∞Directory could be implemented using distributed, redunded an scalable components. In this implementation LMX is not supported which disables license borrowing feature.

Directory distributed Implementation



∞Directory

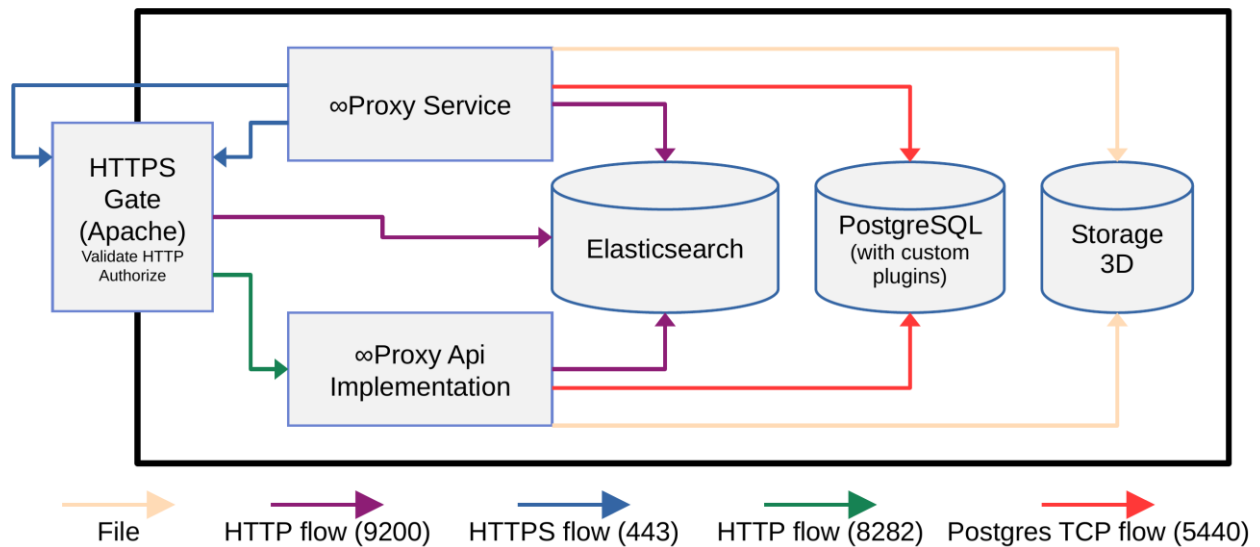
2 - ∞Server/∞Proxy/∞StaticProxy

The ∞Proxy relies on several software components:

- a PostgreSQL server,
- an Elasticsearch server,
- an Apache HTTP server,
- a dedicated service.

The ∞Proxy is implemented as a virtual machine containing all needed components. Scalability and redundancy is obtained by allocating multiple virtual machines. Components of ∞Proxy are not designed to be externalized as they are optimized and customized for optimal performances.

Proxy/Server VM Implementation

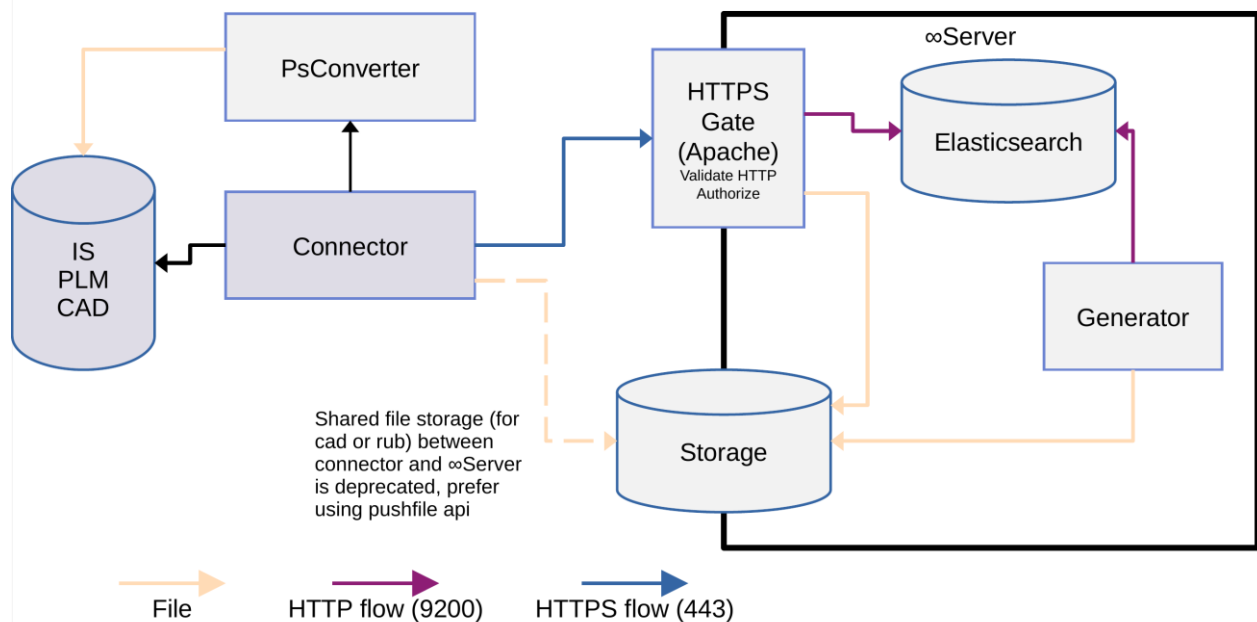


∞Proxy

Databases located on the ∞Proxy are not directly visible to client-side users. Instead, several API are available and only accessible to authenticated users with proper access rights. Cluster-side server-to-server communication relies on HTTPS.

Following diagram shows dataflow between the Connector and the ∞Server. Part of generation load could be scaled by pre-processing all cad files using PsConverter.

Connector and Generator exchanges



∞Proxy

2.1 - ∞Proxy

Under the surveillance of the ∞Directory, a ∞Proxy acts as a proximity relay for DMU broadcasting. It replicates *builds* from ∞Servers or other ∞Proxies. 3D Juump Infinite client applications connect to ∞Proxies in order to browse DMUs. A ∞Proxy is not compulsory to distribute DMU data. For a single point distribution, you can use a ∞Server, which is also a ∞StaticProxy itself, to deliver the DMU to the 3D Juump Infinite client applications.

2.2 - ∞Server

The ∞Server is in charge of the compilation, optimization and packaging of the DMUs. It offers a web interface allowing the enterprise connectors to describe the DMUs (or *projects*). The ∞Server is also a ∞StaticProxy, therefore it publishes *builds* that may be browsed with the 3D Juump Infinite client applications. If you want to replicate the *builds* to multiple sites, you can also install additional ∞Proxies to broadcast the *builds* through your network.

The ∞Server relies on several software components:

- a PostgreSQL server,
- an Elasticsearch server,
- an Apache HTTP server,
- a dedicated service.

Most operations related to the ∞Server administration are controlled by the ∞Directory front-end.

2.3 - ∞StaticProxy

Under the surveillance of the ∞Directory, a ∞StaticProxy acts as a stub server for DMU broadcasting. It is not connected to any streaming data but rather publishes *builds* extracted from DMU files (or *evojuumps*). 3D Juump Infinite client applications connect to ∞StaticProxies just as they would ∞Servers or ∞Proxies.

Access-rights

This chapter explains the access rights policy enforced at the ∞Directory level.

1 - Overview

3D Juump Infinite restricts the access to:

- the different APIs (Directory API, Connector API, Client API...),
- the different nodes of the cluster,
- the DMU builds,
- certain features exploiting the DMUs (export, download...),
- the assets,
- web and native applications,
- the licenses.

Most of these access limitations are configured thanks to tags. For instance, a user will only have access to the builds that match his tags.

2 - Tags

Tags are keywords decorating a resource of 3D Juump Infinite security model:

- builds,
- proxies,
- applications,
- replication link.

A tag is of the form "*keyword*". The *keyword* itself is user-defined.

3 - Access Rights

Access rights are *keyword* regrouped per type. They are applied to consumers of the 3D Juump Infinite security model:

- users,
- teams,
- applications.

Access rights types are :

- **assetadmin** : defines on which projects the user will be asset administrator. Is a list of project ids.
- **connect** : defines on which proxies the user can connect. Is a list of resource tags.
- **download** : defines which builds the user can download in the local DMU manager. Is a list of resource tags.
- **export2d** : defines on which builds the user can generate 2d exports. Is a list of resource tags.
- **export3d** : defines on which builds the user can generate 3d exports. Is a list of resource tags.
- **lmx** : defines if the user is allowed to borrow a license. Mandatory value *allowborrow*.
- **use** : defines which applications (native and/or web) the user is allowed to use. Is a list of resource tags.
- **view** : defines which builds the user is allowed to browse. Is a list of resource tags.

4 - Rules

Access rights management is enforced by applying a set of strict tag-matching rules.

4.1 - Access Rights Inheritance and Validation

During the identification process, the user will inherit tags and access rights will be checked in the following order.

- The user initiates a session for a particular application.
- The user inherits the tags of all the teams he belongs to.
- Access to the application is checked by using the access rights *use*. Access to the application is granted if it has no tags or if all its tags are contained into the user's *use* access right list.
- The user inherits the tags of the application.

- The user lists/opens a build. Only builds whose all tags are contained into the user's *view* access rights and whose hosting proxy has no tags or has all its tags contained into the user's *connect* access rights are granted.

4.2 - User using an application

A user can only use an application if he bears all the matching tags: for each "keyword" tag of the target application, the user must have a matching "**use:keyword**" tag.

4.3 - User connection to a proxy

A user can only connect to a server/proxy if he bears all the matching tags: for each "keyword" tag of the target server/proxy, the user must have a matching "**connect:keyword**" tag.

4.4 - User access to a build

A user can access a feature of a build if he bears all the matching tags.

- Visualisation feature: for each "keyword" tag of the target build, the user must have a matching "**view:keyword**" tag.
- Screenshot feature: for each "keyword" tag of the target build, the user must have a matching "**export2D:keyword**" tag.
- Export feature: for each "keyword" tag of the target build, the user must have a matching "**export3D:keyword**" tag.
- Download feature: for each "keyword" tag of the target build, the user must have a matching "**download:keyword**" tag.

4.5 - User assets administration

A user can administrate assets pertaining to a project only if he bears a matching "_prj_xxx", where _prj_xxx is the project id, into **assetadmin** access right list.

4.6 - User licence borrowing

A user can borrow a licence for offline use only if he bears the "**lmx:allowborrow**" tag.

4.7 - Third party application user access rights

Third party backends may use directory session bearers to limit actions of users that are authenticated on the ∞Directory (using token validation or the directory/api/introspect end point). Using **thirdpartyscopes** access right, it is possible to define application scopes to finely manage access rights. If a user has **myapplicationA** in its **thirdpartyscopes** access rights and if this scope is claimed during the ∞Directory session creation, the access token will contain *myapplicationA* in its *scope* field. Note that the tag value will be percent encoded. See [Third Party application protection](#)

4.8 - Proxy build replication

A proxy can only replicate a build along a server connection if either:

- the connection bears no tag at all (unlimited replication) OR

- the connection bears all the tags of the target build.

All the assets of the projects to which the replicated builds belong will also be synchronized.

Installation

1 - Infinite services installation

This chapter will present how to install an ∞Directory, ∞Server, ∞Proxy or ∞StaticProxy.


1.1 - Requirements


Each 3D Juump Infinite machine must be able to contact (e.g. ping) the ∞Directory host, which is the central point of the 3D Juump Infinite architecture. Each ∞Proxy must be able to contact the ∞Server or ∞Proxy that will provide data to him. Your network administrator must provide you with one or several machine(s) which can be connected (in the network sense).


Before starting installation you should retrieve:

- The software delivery package containing Windows installers, install scripts, configuration files and this documentation
- At least one machine to host your ∞Directory / ∞Server
- **Administrator** rights on all machines
- Python 2.7 or 3.6
- For ∞Directory, an html 5 compliant browser
- For linux, an internet access to REAL FUSIO apt repository from all machines
- Optionally (but strongly recommended), a SSL certificate, such a certificate may be built

- Optionally, setup or use an existing LMX server, configured with REAL FUSIO vendor extension (such a server can be installed on the same host than the ∞Directory [LMX])

 **YOU CANNOT INSTALL** an ∞Server and an ∞Proxy or an ∞StaticProxy on the same machine.

 All logins/passwords must be limited to alphanumerical characters and these separators: space ' ', minus '-', underscore '_' and point '.'¹⁰.

 You might require to temporary disable your antivirus during the installation procedure. We need to modify the windows firewall rules during installation. Some antivirus software (e.g. McAfee) may detect this as a potential threat.

Notes: All credentials will be stored in an encrypted form in order to ensure that only a trusted user of the machine can access this information.

The ∞Directory, ∞Server, ∞Proxy and ∞StaticProxy software run on any of the following operating systems with an **IPv6 stack**:

- Microsoft Windows 7 and above, 64-bit version
- Linux Debian 11 bullseye, AMD64
- Linux Ubuntu 20.04 LTS (focal) AMD64

Minimum hardware requirements are:

- Quad-core processor (support of POPCNT x86 instruction is mandatory, support of CRC32 x86 instruction is recommended)
- For the ∞Directory, ∞Proxy and ∞StaticProxy : 8GB of RAM
- For the ∞Server : 4GB of RAM by CPU Core (A Quad-core processor should have at least 16 GB of RAM).
- 1GB disk space for binaries + sufficient disk space for data (depending on your data sources)
- High-speed hard drive disk or solid-state drive highly recommended

1.2 - Third party service version

3DJuump Infinite installs and uses third party services. Here is a table that lists expected version for these components.

Third Party Software	∞Directory	∞Server / ∞Proxy / ∞StaticProxy
LMX Server	4.9.15	-
PostgreSQL	13 (13.7+)	13 (13.7+)
ElasticSearch	-	6 (6.8.23+)
Java JRE Server	-	8 (8u222+)
Apache	2.4 (2.4.53+)	2.4 (2.4.53+)

¹⁰ For information, the matching regex is "[a-zA-Z0-9_-.]+".

1.3 - Install scenarios

Depending on your needs we provide the following install scripts for the given install scenarios:

- `directory` : will install an ∞Directory on the machine.
- `proxy_or_server` : will install an ∞Proxy, ∞StaticProxy or ∞Server on the machine.
- `directory_and_proxy_or_server` : will install an ∞Directory and an ∞Proxy, ∞StaticProxy or ∞Server on the machine.

The folder `install_form` of the software delivery package contains the different install scripts for each scenario (e.g `directory_install.py`, `proxy_or_server.py`, etc ...) and a corresponding form (e.g `form_directory.yaml`, `form_proxy_or_server.yaml`, etc ...) that lists the installation settings. First edit the corresponding `form_xxxxxxx.yaml` and fill all mandatory entries. You are free to update entries that are pre-filled. Optional entries should be left empty except if you have specific configuration needs. You will find a description of each entry in `doc_XXXX.html` next to each form. Once you have finished editing the `yaml` file, open a terminal with **Administrator** rights and run the corresponding `xxxxxxx_install.py` python script. Then follow the instructions.

All install scripts support the following options:

- `-f` : force answer yes to all user input
- `-accept-eula` : automatically accept EULA (end-user license agreement)

Notes: You may provide your own SSL certificate and private key (in `x509` format without password protection) for the TLS communication. This certificate should match the declared public url in the `form_xxxxxxx.yaml` file. You may then proceed with the installation and override the autogenerated certificate and private key with your own files.

1.4 - Test the ∞Directory front-end web page

For any scenario involving the installation of an ∞Directory, once the installation is finished, the ∞Directory front-end administration page can be accessed through `https://directory_name:${apache_https_port}/directory` (see [Login in the web administration interface](#)).



The web administration interface front page cannot be displayed? Check if the ∞Directory service has started and is approved by your network firewall rules.

2 - LMX (optional, for borrow support)

2.1 - Run a LM-X server (for borrow support)

An LM-X server is required only to get borrow support within the local DMU Manager.

If you do not already have an available LM-X server (version 4.9.15), please run the corresponding installer (provided in the install package) and refer to the LM-X documentation. Provide the REAL FUSIO vendor extension file (`LibLmxvendor.so` on Linux or `LibLmxvendor.dll` on Windows) when required during the installation.

Note: it is not necessary that the LM-X server run on the same physical computer as the ∞Directory as long as it is visible (in a network sense) from both to the ∞Directory and the 3D Juump Infinite client application stations.

Remember the LM-X *license path* (see LM-X documentation) corresponding to your LM-X server (or LM-X cluster), usually something similar to "6200@my-lmx-server" (`${Lmx_server}`).

2.1.1 - Install the LMX borrow license file

Copy the license file (*.Lic) provided by REAL FUSIO to the LM-X server folder.

2.1.2 - Extract the ∞Directory keys

1. Open the license file in a text editor.
2. Find the lines starting with `FEATURE DIRECTORY. FEATURE DIRECTORY-PROD-5347F99D-5479-4071-A057-9351DCE3A9AF (VENDOR=REALFUSIO COUNT=1 VERSION=... ..`

Each such line corresponds to an instance of ∞Directory. In the given example, the key is `DIRECTORY-PROD-5347F99D-5479-4071-A057-9351DCE3A9AF` and it corresponds to a *production* ∞Directory (hence the *-PROD-*). Depending on your contract, you may find several ∞Directory keys in the license file:

- `DIRECTORY-PROD-...` is a key for a *production* ∞Directory.
- `DIRECTORY-EVAL-...` is a key for an *evaluation* ∞Directory.
- `DIRECTORY-TEST-...` is a key for a *test* ∞Directory.

Remember the ∞Directory key matching the type of the ∞Directory being installed (`${Lmx_feature}`).

3 - OpenId Connect

User authentication is deferred by the ∞Directory to a third party server using the [OpenId Connect code flow protocol](#). The OpenId configuration is specified during installation in `form_*.yaml`. Detailed descriptions of each field are available in `'/manual/api_and_service_conf/directory_conf.html'`

3.1 - Configuration

To properly configure OpenId delegation you have to request the creation of an *Application* to your ID provider. This *Application* redirect uri will be `https://host:443[/prefix]/directory/api/directorysession/onauthenticated`. In exchange you should obtain:

- A configuration url [OpenID Provider configuration url](#)
- An application id
- An application secret
- Whether PKCE should be enabled or not
- Which kind of JWT signing algorithm will be used

3.2 - User identification

When opening an ∞Directory session, the user is identified using the OpenId Connect code flow protocol. An `id_token` is retrieved. Optionally an `access_token` and a `refresh_token` could be requested depending on the `use_oidc_access_token` setting.

3.3 - Id token usage

The Id token delivered by the OpenId server is used to identify the user. The following standard fields are extracted from the `id_token`: `sub`, `name`, `given_name`, `family_name`, `middle_name`, `nickname`, `preferred_username`, `profile`, `picture`, `email`, `email_verified`, `zoneinfo`, `locale`, `phone_number`, `phone_number_verified`, `address`, `updated_at`.

Those fields are provided through the `directory/api/manage/users` api and are used to compute the user's display name. If delivered, the `id_token` contains non standard fields that can be mapped to standard fields by using the `id_token_alias` setting.

3.4 - Authentication webhook

An optional authentication webhook could be configured using the `authentication_webhook` setting. This webhook will be called before authorizing each new directory session. It could be used to update user credentials using the directory api. The webhook should expect an HTTP POST request with a json payload ("Content-Type: application/json"):

```
{
  "sub": "...",
  "access_token": "...",
  "challenge": "..."
}
```

- The `sub` field will contain the OpenId Connect sub id.
- The `access_token` field will contain the access token delivered by the OpenId Connect server.
- The `challenge` field specifies the challenge that must be returned in the webhook response.

To authorize the user, the webhook should respond with an HTTP response code 200 and a json payload ("Content-Type: application/json") :

```
{
  "challenge": "...",
  "authorized": "true"
}
```

3.5 - Access Token usage

It is possible to protect the ∞Directory and ∞Proxy APIs using `access_tokens` delivered by the OpenId server. This feature is enabled by `use_oidc_access_token` setting. When enabled OpenId server should deliver a [JWT](#) `access_token` and a `refresh_token`. The `access_token` will be sent to the client, the `refresh_token` will be kept on the ∞Directory to renew the `access_token`. Access token scopes can be customized using the `additional_scopes` setting. The first `access_token` will

be retrieved using *additional_scopes/primo_token* and will only be sent to the authentication webhook. Client access_tokens will be retrieved using *additional_scopes/client_token*. The *oidc_access_token_aud* setting can be used to specify which audience should be expected in access_tokens.

If *use_oidc_access_token* is disabled, ∞Directory and ∞Proxy api will be protected using internal tokens.

3.6 - Non standard query parameters

Custom query paremeters can be added to OpenId endpoints calls using *additional_query_parameters*.

3.7 - Example

Configuration example for Azure AD

```
openidconnect:
  allowed_jwt_alg:
    - RS256
  client_id: MyApplicationId
  client_secret: MyApplicationSecret
  configuration_endpoint: https://login.microsoftonline.com/xxxxxxx/v2.0/.well-known/openid-configuration
  use_PKCE: true
  verify_ssl_peer: true
  use_oidc_access_token: true
```

Configuration example for Auth0

```
openidconnect:
  additional_query_parameters:
    authorization_endpoint:
      audience: https://mytestapi
  allowed_jwt_alg:
    - HS256
    - RS256
  additional_scopes:
    primo_token: mytestscope offline_access
    client_token: ''
  client_id: MyApplicationId
  client_secret: MyApplicationSecret
  hmac_secret: MyApplicationSecret
  configuration_endpoint: https://xxxxxx.eu.auth0.com/.well-known/openid-configuration
  use_PKCE: true
  verify_ssl_peer: true
  use_oidc_access_token: true
```

Administration

This chapter provides hints for the following administrative tasks:

- manage 3D Juump Infinite through its web application interface
- enumerates all the services started and opened and TCP/IP ports to assist you to properly install 3D Juump Infinite in conformance with your enterprise security policy
- activity script for billing purpose

1 - Web application well known urls

3D Juump Infinite directory serve a group of well known urls. *{prefix}* is defined at installation, by default it will be empty.

Url	Description
{prefix}/directory/home	home page, listing application available to a logged user
{prefix}/directory/front-end	directory administration page
{prefix}/directory/webclient	build-in web client
{prefix}/directory/webapi/infiniteapi3.3.js	web api javascript file
{prefix}/directory/webapi/doc/index.html	web api documentation
/	redirect to {prefix}/directory/home/
/directory	redirect to {prefix}/directory/frontend/

/webcliend	redirect to {prefix}/directory/webclient/
{prefix}/directory/webapi	redirect to {prefix}/directory/webapi/doc/index.html

2 - Administation web application interface

First, make sure that the ∞Directory has been installed properly and is running along with its dependent back-end services. The ∞Directory provides a web application interface that lets the administrator manage the 3D Juump Infinite network, data and user rights. This web application is accessible through the ∞Directory web server, by browsing the corresponding URL:

[https://directory_name:\\${apache_https_port}/directory/frontend](https://directory_name:${apache_https_port}/directory/frontend)



Please display the web administration interface with a supported browser: Microsoft Edge, Google Chrome or Mozilla Firefox.

2.1 - Login

The *Login* page prompts the administrator to prove his identity through the defined delegation. If Single Sign-On (SSO) is supported by your *authenticator*, the login page is skipped.

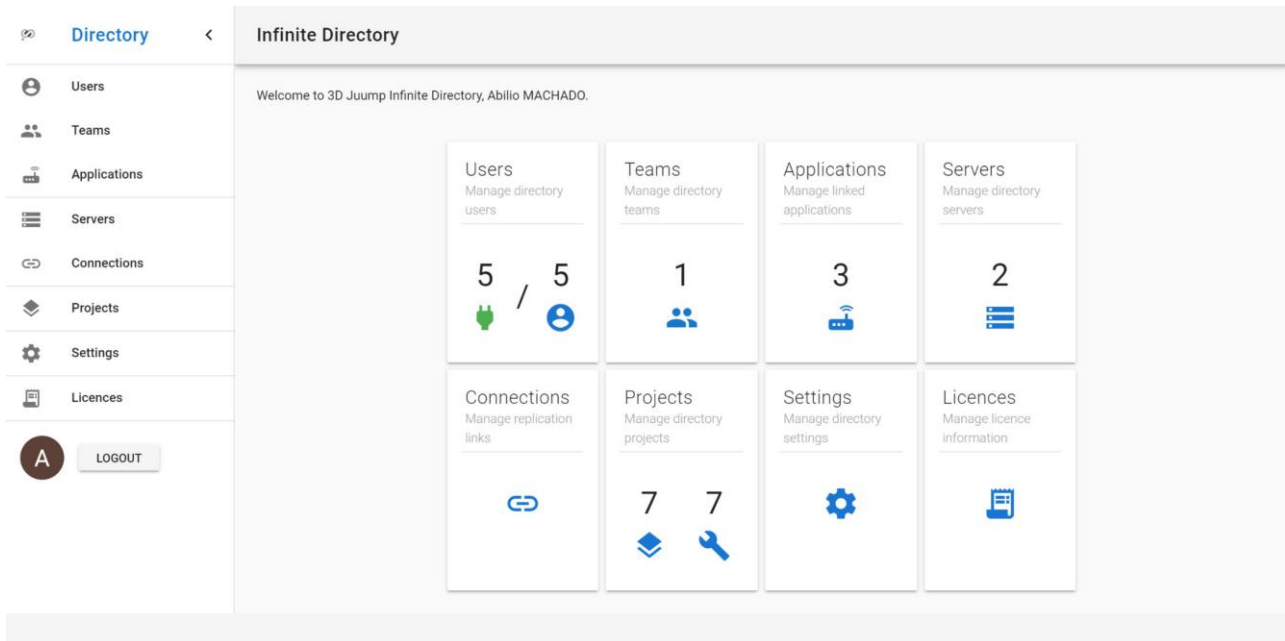
Login

Only users with administrator privilege can successfully login to the front-end. One exception is the superuser login: it is always possible to fallback to a special login procedure relying on the ∞Directory PostgreSQL superuser credentials. In particular, upon prime installation, the

∞Directory has no registered user, thus no registered administrator: use the fallback superuser login procedure to initiate the ∞Directory user database with its first administrator.

2.2 - Dashboard

Once logged in, the browser proceeds to a *Dashboard* summarizing the current state of 3D Juump Infinite.



Dashboard

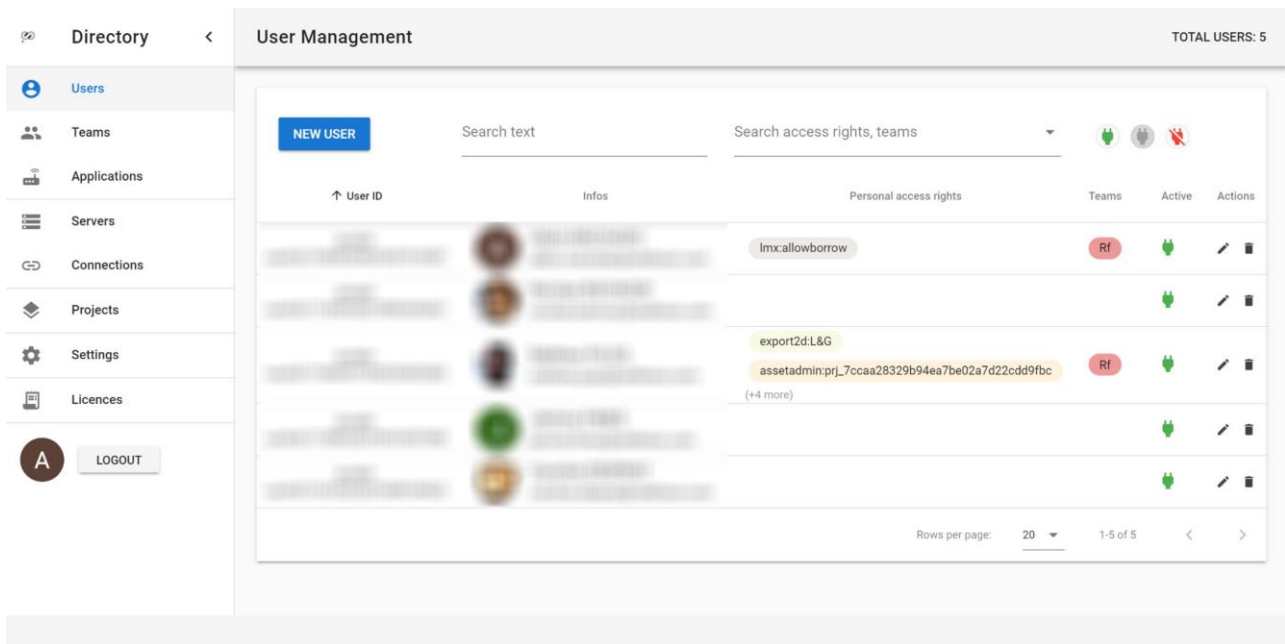
It lists several statistics about 3D Juump Infinite users, teams, projects and servers.

A menu bar, located at the left of the *Dashboard*, gives access to dedicated administration panels. Note the top-left infinite icon: it is a shortcut for a quick access to the *Dashboard*.

2.3 - Team Management

2.3.1 - Users

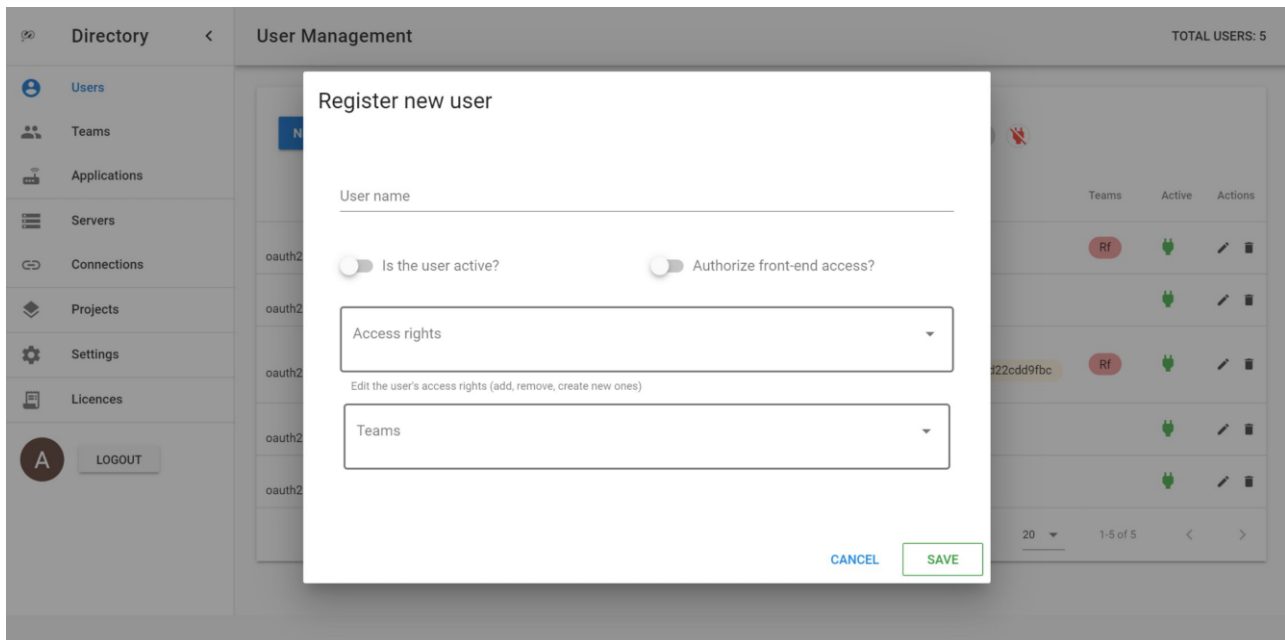
The *User Management* page lets the administrator create, edit and delete users. It also lists the registered users.



Users

2.3.1.1 - Create Users

When the administrator needs to create a new user, he clicks on the [New User](#) button (top-left) and fills in the user id for this user (the one delivered by the *authenticator*). It is also possible to activate the user, to grant him administrator privilege (the ability to access the *front-end*), to decorate him with tags or to inscribe him in teams.

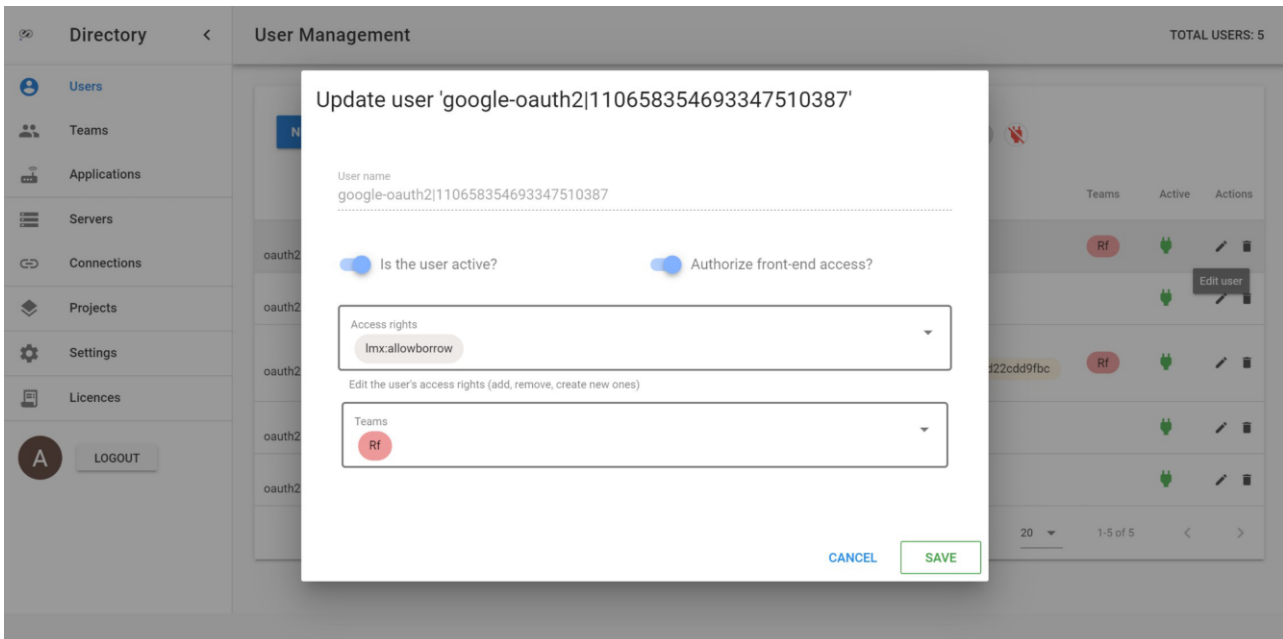


Create User

Another way to create users is to have them login from the 3D Juump Infinite client application. Once properly authenticated, the user id is automatically registered in the ∞Directory users database, though the corresponding user remains inactive, has no privilege, nor tags, nor teams.

2.3.1.2 - Edit Users

The administrator can edit a user by clicking on the pencil icon that appears on the corresponding line.



Edit User

Of course, it is impossible to change the user id of a given user as it links him to his *authenticator*-provided identity.

2.3.1.3 - Delete Users

To delete an existing user, the administrator can click on the trash-can icon that appears on the corresponding line.

2.3.1.4 - Batch User Creation

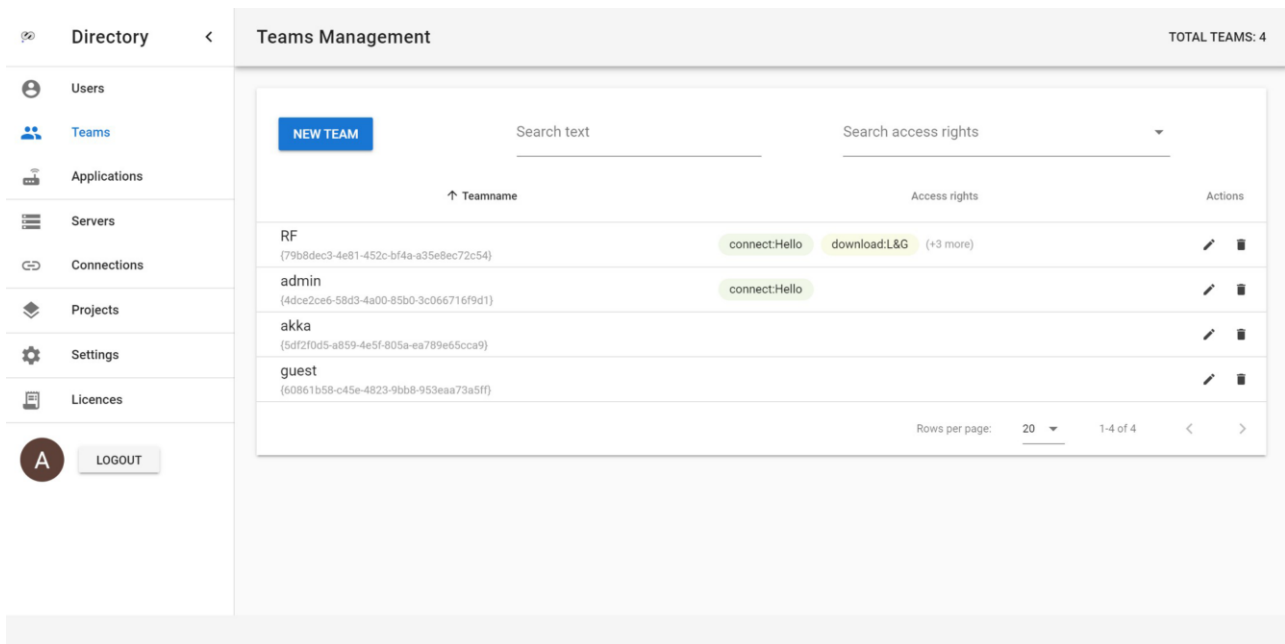
Batch user creation is not part of the *front-end* anymore and is only available through the *Directory API*.

2.3.1.5 - Send messages

It is not possible to send a message to selected users anymore: to address messages to specific users, we suggest you rather use emails.

2.3.2 - Teams

The *Team Management* page lets the administrator create, edit and delete teams. It also lists the created teams.

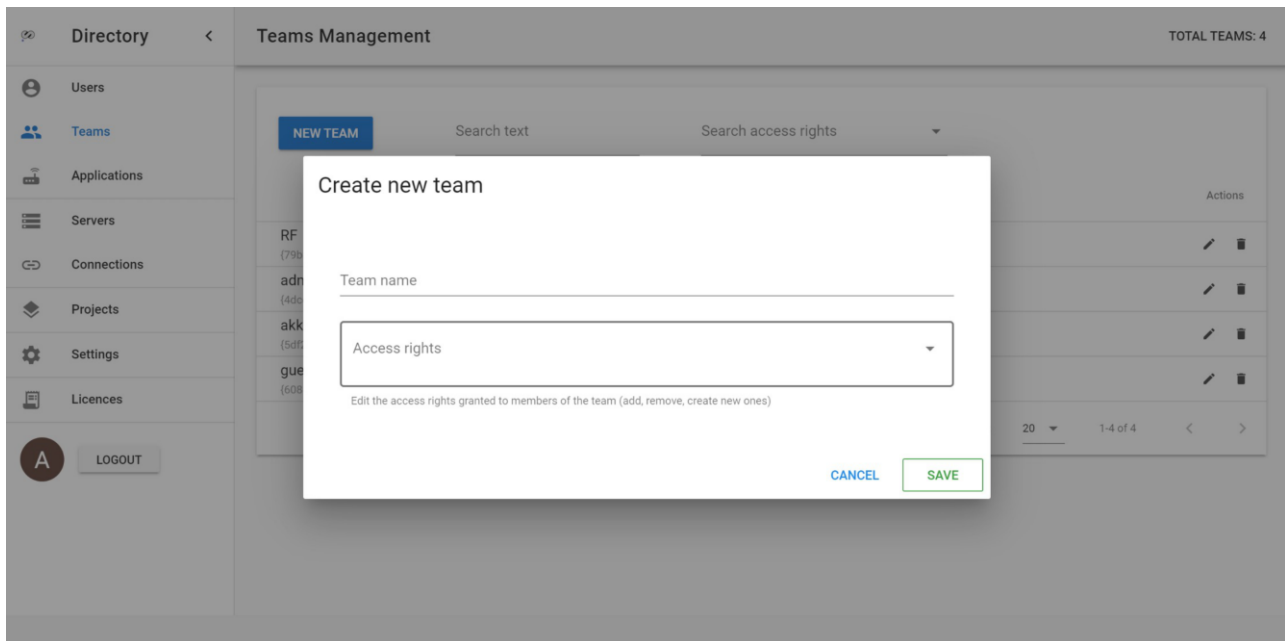


Teams

Teams are useful to quickly set tags to a group of users.

2.3.2.1 - Create Teams

When the administrator needs to create a new team, he clicks on the *New Team* button (top-left), fills in the (unique) name for this team and sets its tags.

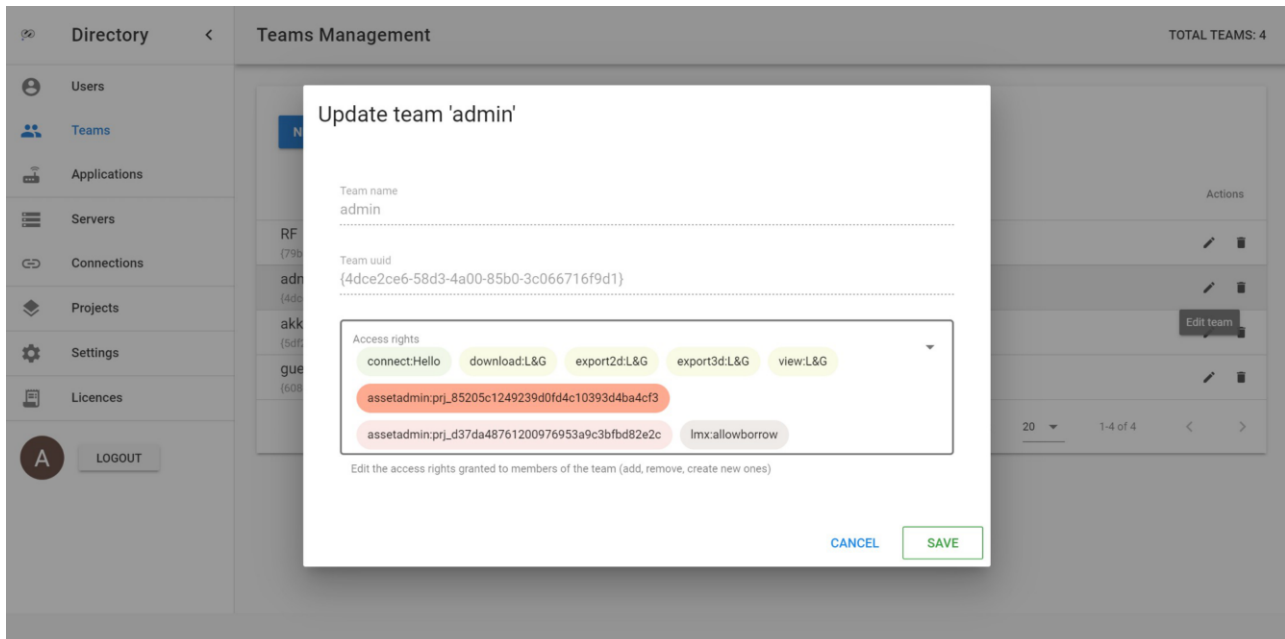


Create Team

Note: it is not possible to change a team's name once it is created.

2.3.2.2 - Edit Teams

The administrator can edit a team's tags directly by clicking on the pencil icon that appears on the corresponding line. It is possible to change the tags associated to the team.



Edit Team

2.3.2.3 - Deleting Teams

To delete an existing team, the administrator can click on the trash-can icon that appears on the corresponding line.

2.3.2.4 - Send messages

It is not possible to send a message to teams anymore.

2.4 - Network

2.4.1 - Applications

The *Applications Management* page lets the administrator register new applications to the 3D Juump Infinite network of this ∞Directory. It also lists the registered applications.

REGISTERED APPLICATIONS: 3

REGISTER APPLICATION

Search text

Search tags, accessrights

Application name	Description	Tags	Access rights	Actions
3D Juump Infinite Directory	3D Juump Infinite Directory front-end administration page			
3D Juump Infinite Native Client	3D Juump Infinite Native client application			
ADSQ				

Rows per page: 20 1-3 of 3

Applications

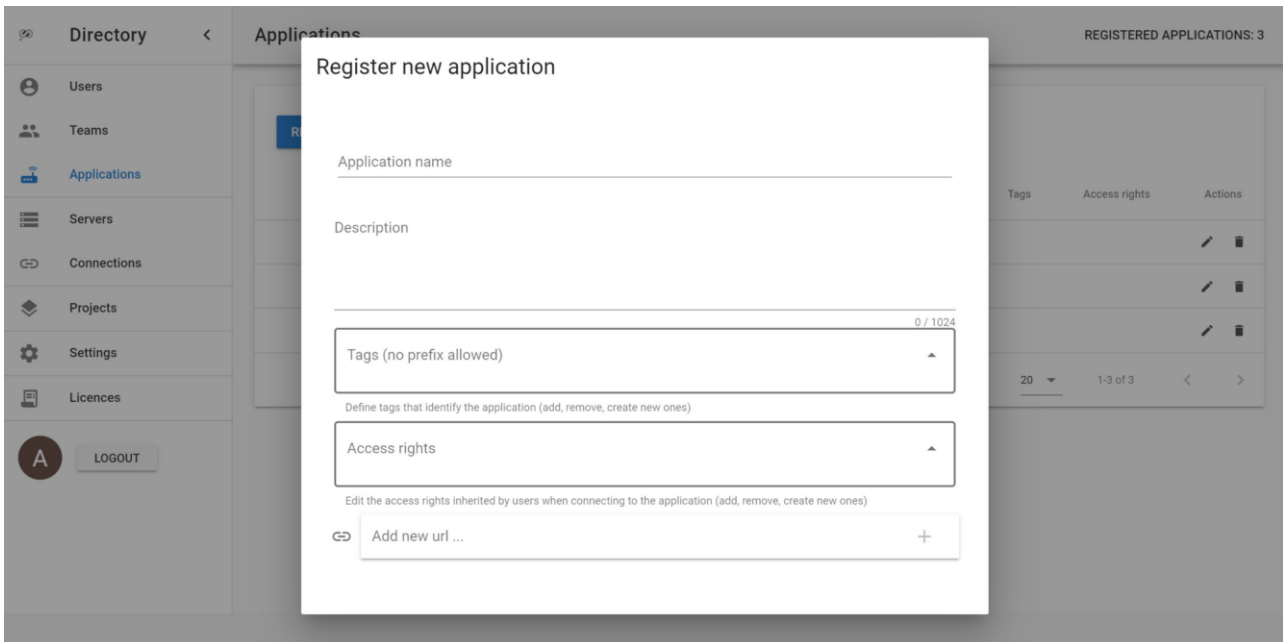
2.4.1.1 - Application Info

The list displays one line per application with several pieces of information (from left to right):

- The left-most column contains the application name.
- The description of the application
- The *Tags* column displays the tags of this application, that is to say the tags that a user must match to use to this application.
- The *Access rights* column specifies among other things the rights to the different properties of the builds for the users of the application.
- The *Actions* column gives access to several actions listed below.

2.4.1.2 - Register Applications

The administrator can register a new application by clicking on the *REGISTER APPLICATION* button (top-left) and filling in the form with:



Registering Applications

- The name of the application (for human-readable name displayed in the ∞Directory).
- A short description of the application
- Tags decorate the application and effectively limit users' use of the application.
- The *Access rights* specify, among other things, the rights to the various project properties for the users of the application.
- A list of valid urls that are used by the application. (e.g.: the 3D Juump Infinite Native Client needs something like

<https://mydirectory:443/directory/api/directorysession/onauthenticationdone>)

2.4.1.3 - Update Applications

To update an existing application, the administrator can click on the pencil icon in the *Actions* column.

Except for the application name, you can change all other properties of the application.

2.4.1.4 - Delete Applications

To delete an existing application, the administrator can click on the trash-can icon that appears on the corresponding line.

2.4.2 - Servers

The *Servers Management* page lets the administrator manage servers (∞Directory ∞Server, ∞Proxy or ∞StaticProxy) to the 3D Juump Infinite network of this ∞Directory. It also lists the registered servers. As you might guess, it requires some installed ∞Servers; please go to ∞Server installation section if no ∞Server has been installed yet.

Directory < Servers Management TOTAL SERVERS: 2

Search text Search tags

Info	Label	URL	PG Port	Tags	Active connections	Accept new connections	Actions
	Directory	https://172.25.162.5	5440		0		
	172.25.162.5_0_proxy	https://172.25.162.5/proxy	5440	Hello	0		

Rows per page: 20 1-2 of 2

LOGOUT

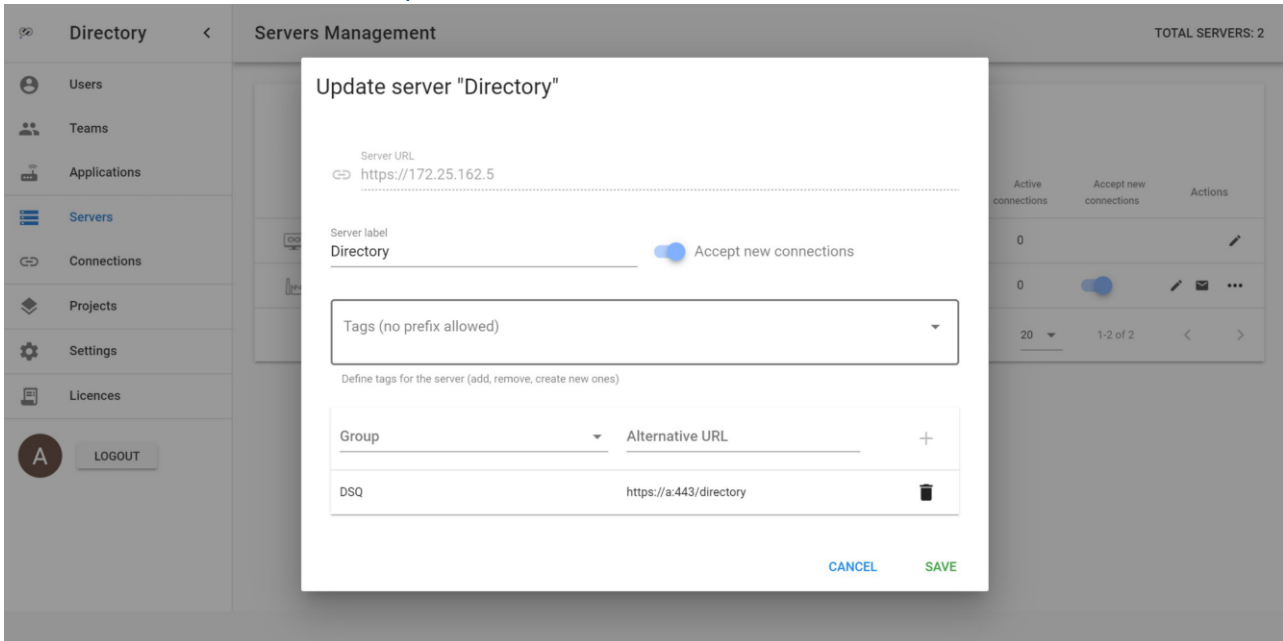
Servers

2.4.2.1 - Server Info

The list displays one line per server with several pieces of information (from left to right):

- The left-most column contains an icon representing the type of server (either a or a).
- A status icon indicates whether the server is properly running or not. On mouse-over, it also reports information regarding the server's sub-services status and versions.
- The *Label* column contains the name of the server.
- The *URL* column contains the host address of the given *Server*/*Proxy*.
- The *PG Port* columns contain the PostgreSQL port of the server.
- The *Tags* column displays the tags of this server, that is to say the tags that a user must match to connect to this server.
- The *Active connections* column displays the number of sessions currently opened to this server.
- The *Accept new connections* column displays if the given *Server*/*Proxy* may accept user connections.
- The *Actions* column gives access to several actions listed below.

2.4.2.2 - Update Servers



Update Server

the server registers itself with the directory. if the server is not visible in the list check the server configuration. To update an existing server, the administrator can click on the pencil icon in the *Actions* column.

The administrator can modify:

- The server label (for human-readable label displayed in the ∞Directory).
- If the ∞Server/∞Proxy can accept user connections.
- The tags decorating the server and effectively limiting the user's connections.

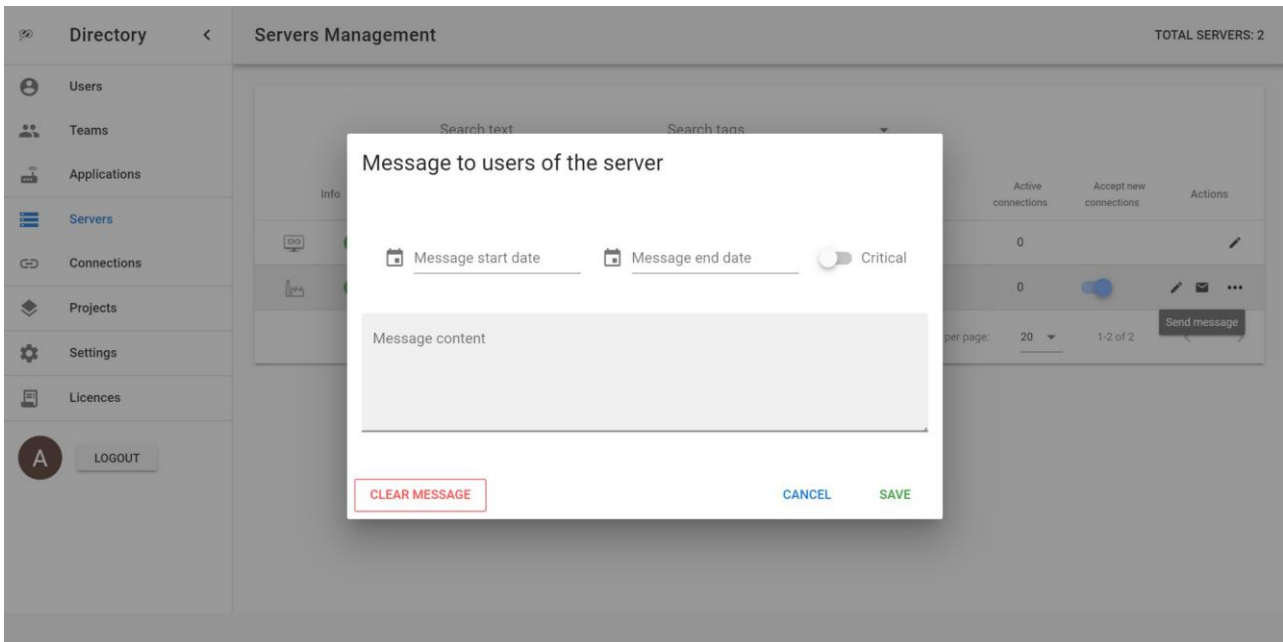
The administrator can add:

- Group and alternative URL for this group

The alternative URL will be used for communication between servers of the same group.

2.4.2.3 - Send message

To send a message to users on a server, the administrator can click on the mail icon in the *Actions* column.

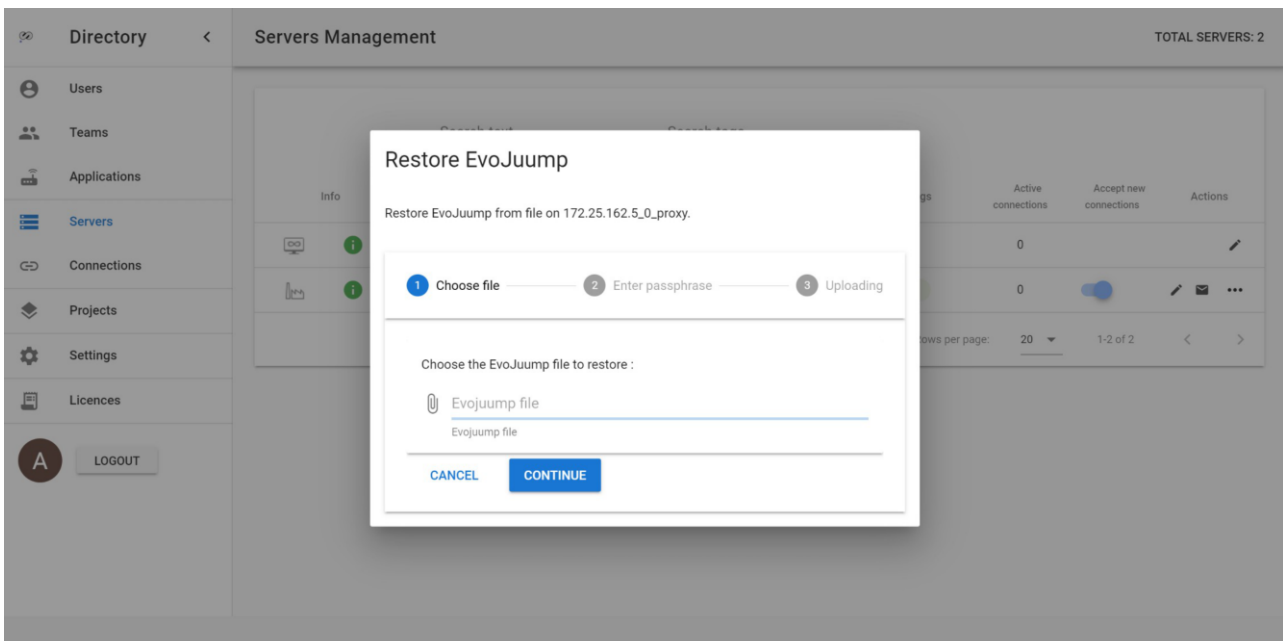


Send Message

The corresponding form lets the administrator define the message displayed on the server's billboard (when it is visible, what is its content, whether it is critical or standard...). The server's billboard is automatically shown to users connected or connecting to the server. The administrator can also clear the server's billboard.

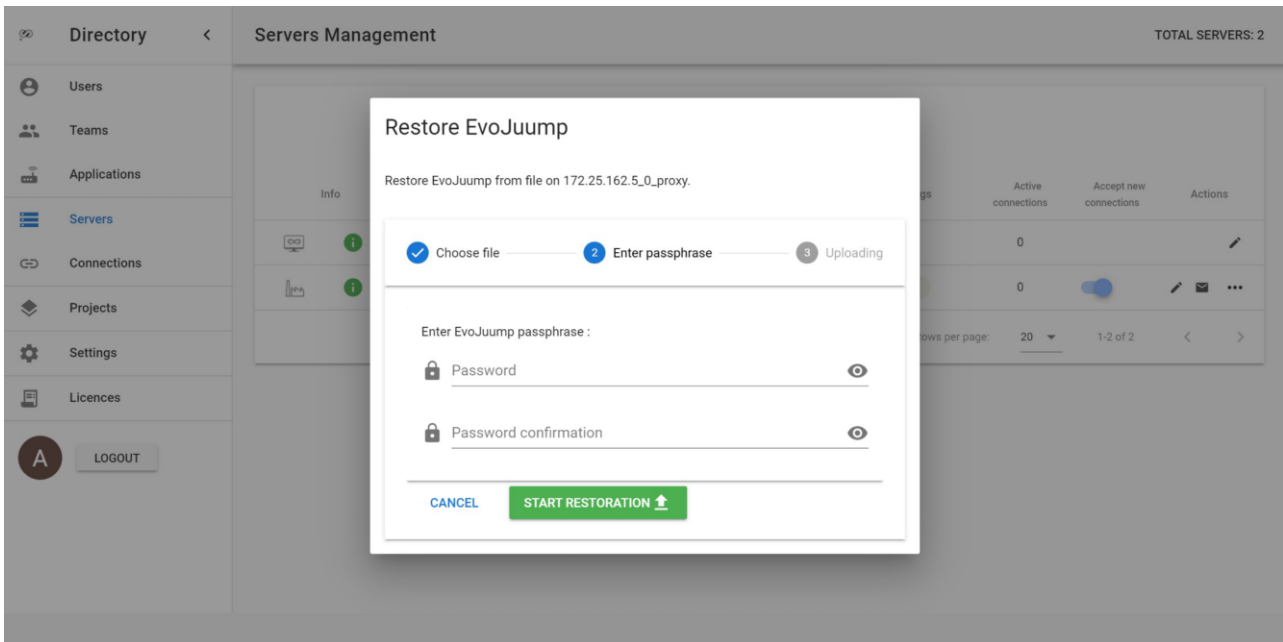
2.4.2.4 - Restore builds

The administrator can also restore a build previously packed. It is feasible on any ∞Server or ∞StaticProxy by selecting the corresponding action in the *Actions* column.



Choose file

First, select the evojump file from your drive.



Enter passphrase

Then provide the passphrase (if any).

Fill in the required information and proceed to the build installation.

2.4.2.5 - Clean-up Proxy

It is possible to clean-up unreferenced documents from a server/proxy by clicking on the corresponding action in the *Actions* column.

2.4.2.6 - Revoke all users

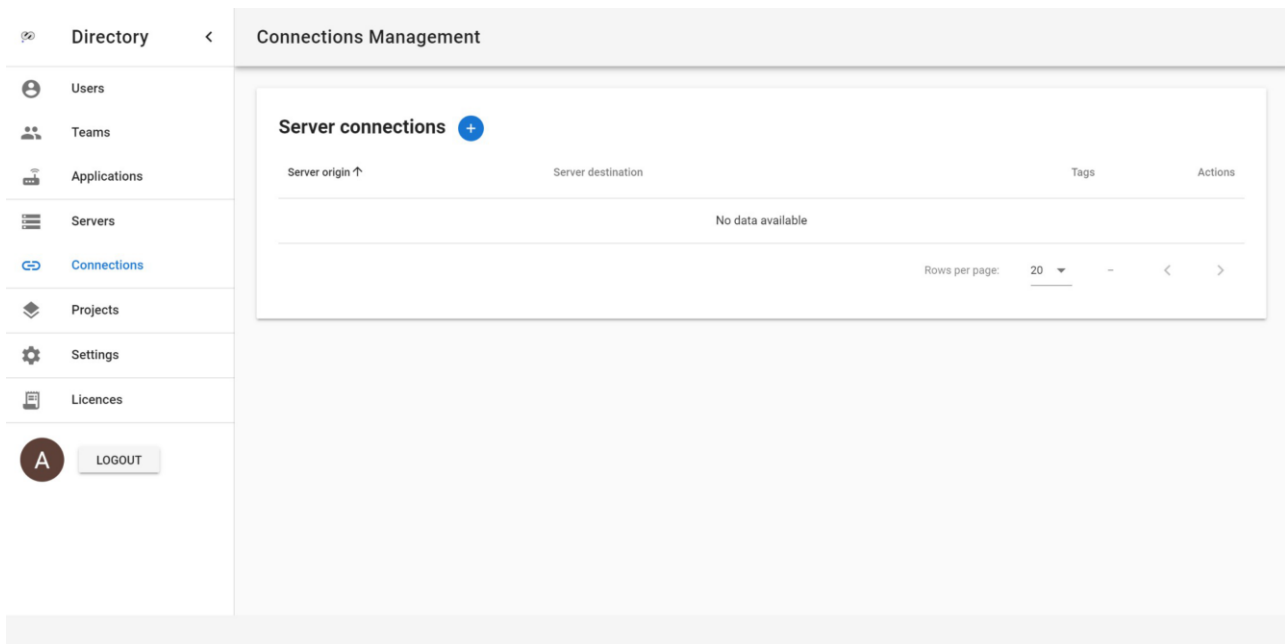
The administrator can forcefully revoke all connected users from a server by clicking on the corresponding action in the *Actions* column.

2.4.2.7 - Delete Servers

The administrator can delete an existing server by clicking on the corresponding action in the *Actions* column. This is restricted to servers that are currently offline.

2.4.3 - Connections

The *Connections Management* page lets the administrator define the [DMU-flow](#), the manner in which 3D Juump Infinite connects ∞Proxies and ∞Servers to broadcast the DMU to the 3D Juump Infinite client applications. It also provides a list of the existing connections.



Connections

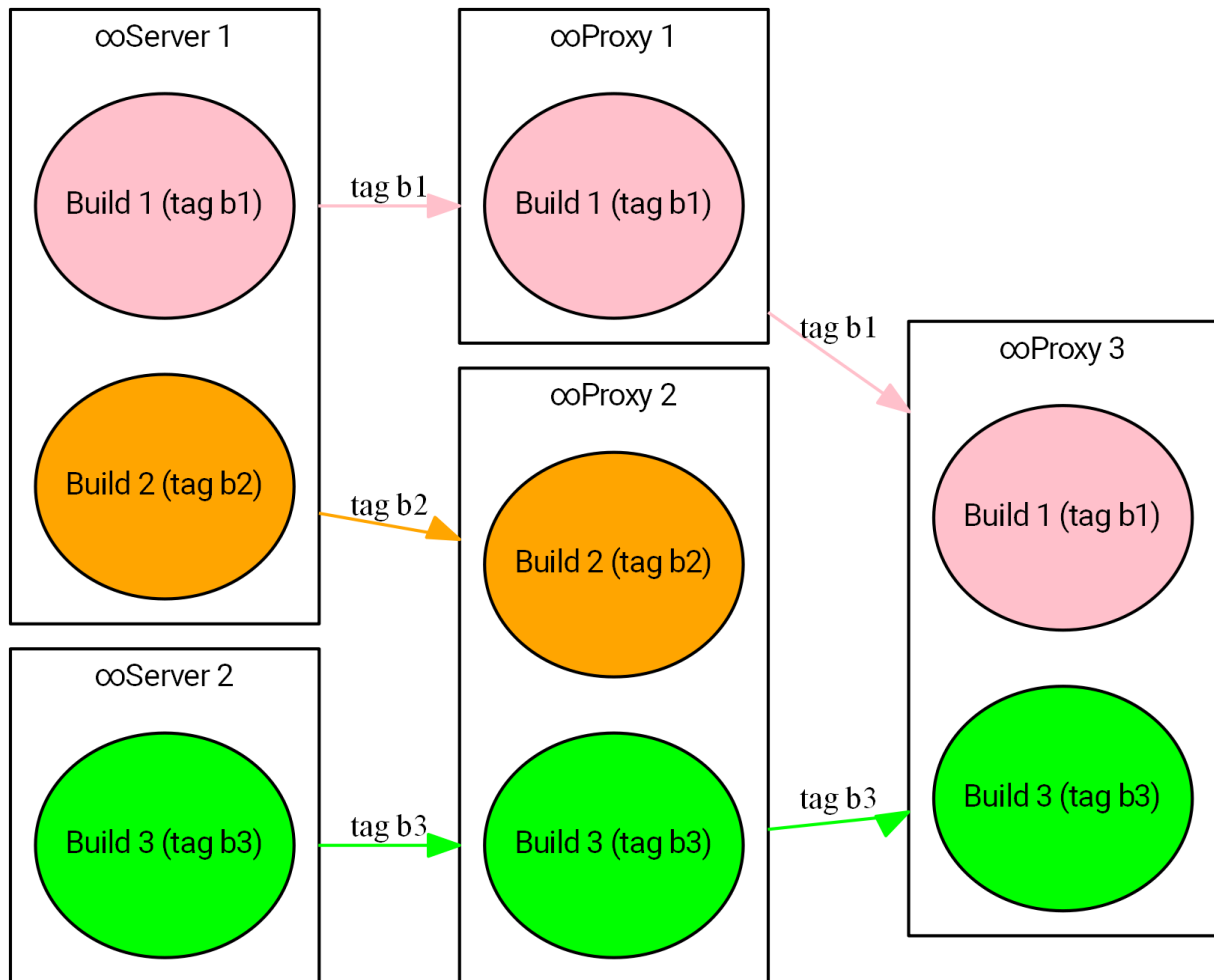
2.4.3.1 - Server connections

Each ∞Proxy replicates all the *builds* and *assets* from its *sources* (the ∞Servers or ∞Proxies that it takes its data from) given that the *connection* matches the tags of the *builds*.

In the connections list, from left to right:

- The *Server origin* column indicates the source ∞Server or ∞Proxy.
- The *Server destination* column indicates the destination ∞Proxy.
- The *Tags* column indicates the limitations to builds replication. If no tags are set, all builds are replicated. Else, only the builds matching the tags are replicated.

All the assets of the projects the matching builds belong to will also be synchronized.



Sources

In the example, two ∞ Servers generate three projects and are replicated by three ∞ Proxies:

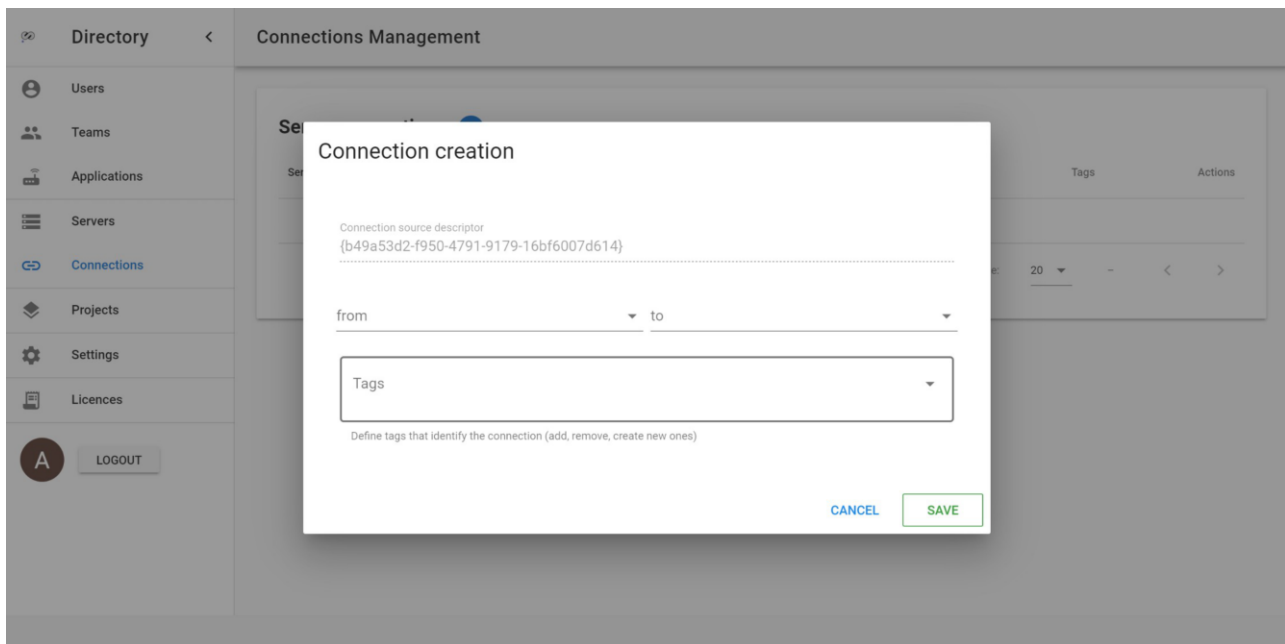
- ∞ Server 1 generates two builds tagged resp. *b1* and *b2*.
- ∞ Server 2 generates one build tagged *b3*.
- ∞ Proxy 1 has ∞ Server 1 as its only source and its connection is tagged *b1* only.
- ∞ Proxy 2 has two connections (with ∞ Server 1 and ∞ Server 2) tagged resp. *b2* and *b3*.
- ∞ Proxy 3 has two connections (with ∞ Proxy 1 and ∞ Proxy 2) tagged resp. *b1* and *b3*.

All the assets of the projects to which the replicated builds belong will also be synchronized.

2.4.3.2 - Add connection

The administrator can add a new connection by clicking on the **+** button and fill in the form with:

- The source ∞ Server or ∞ Proxy.
- The destination ∞ Proxy.
- The tags to limit replication.



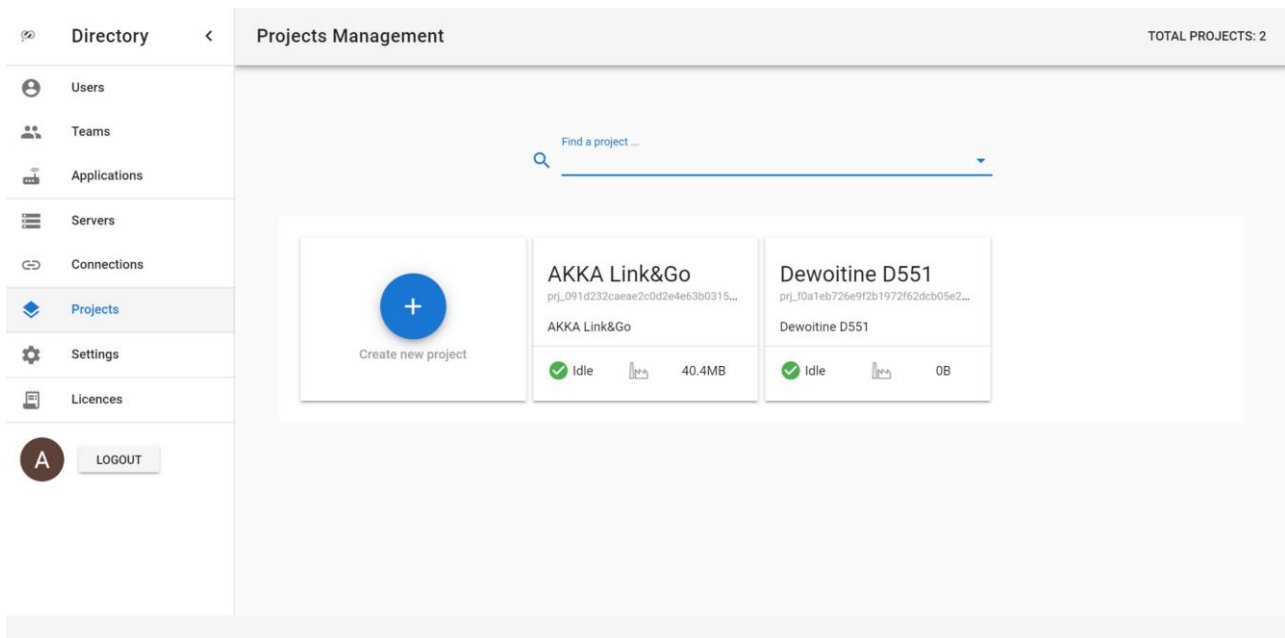
Adding Sources

2.4.3.3 - Delete connection

The administrator can remove a connection by clicking on the trash icon.

2.5 - Projects

The *Projects* page lets the administrator create new projects, check the status of existing projects and list existing builds.



Projects

2.5.1 - Create Projects

When the administrator needs to create a new project, he clicks on the **+** button. Then, a form pops up and requires entering the following information:

- The human readable name of the project.
- The ∞ Server which will host the project.
- The project's unit.

Note: the creation of a project may take several minutes.

The newly created project will appear in the project selection combo-box. After few minutes, please refresh the web page to display project creation status.

The screenshot shows the 'Create project' modal form. The form has the following fields and options:

- Project name:** A text input field.
- Generator:** A dropdown menu.
- Unit:** A dropdown menu showing 'Project ID' and the value 'prj_f3d2102d9ac616b80031a1b8a8000301'.
- Or, restore a project:** A checkbox.
- Please, upload a project description:** A text area with a placeholder 'Please, upload a project description.'
- Buttons:** 'CANCEL' (blue) and 'SEND EVENT' (green).

The background shows the 'Projects Management' page with a sidebar containing 'Users', 'Teams', 'Applications', 'Servers', 'Connections', 'Projects', 'Settings', and 'Licences'. The 'Projects' tab is selected. The top right corner of the page indicates 'TOTAL PROJECTS: 2'.

Project Creation



Please refresh the web page to update the creation status display.



This project unique identifier might be an useful piece of information for the connector development.

Note: it is now possible to create a project from its *project descriptor*. It is required for the backup-restore procedure.

2.5.2 - Check a Project Status

The *Project Management* page gives an overview of all projects status. For more detailed information, click on the project's tile.

The screenshot shows the 'Project Link&Go' interface. On the left is a sidebar with navigation links: Directory, Users, Teams, Applications, Servers, Connections, Projects (selected), Settings, and Licences. At the bottom of the sidebar is a user profile icon 'A' and a 'LOGOUT' button. The main content area shows the project name 'Project Link&Go' with a long ID and 'TOTAL BUILDS: 1'. Below this is a 'Status: Idle' indicator with a green checkmark and a 'BUILD' button. A 'Build comment' box shows a comment ID and timestamp '02 AUGUST 2021 12:10 UTC'. Below the comment is a 'PROXIES' tab with a sub-tab 'EVOJUUMP' and a green checkmark. A table lists proxies with columns: Role, Label, Status, and Action. One proxy is listed with Role '172.25.162.5_0_proxy', Status 'HTTP' with a green checkmark, and an action icon.

Project page

A status bar displays the current status of the project, either:

- **Idle**: the project is in idle state, waiting for new data.
- **Updating**: the project is currently being fed with new data.
- **Building**: the project is currently compiling a new *build*.
- **Packing**: the project is currently packing a *build* for export.

The status can also denote an error in the update process. See *Project status* section of the integration manual for more informations.

The actions available in the project menu are described below.

2.5.3 - Request a Build

In the project menu, the **Build** action commands the ∞Server to generate a new *build* as soon as the project is *buildable* (i.e. it is in *idle* state and the connector has declared that the database is coherent).



Please refresh the web page to update the build status.

2.5.4 - Synchronize metadata

In the project menu, the **Synchronize metadata** action commands the ∞Server to update the metadata of existing *build* with new data from the *Connector*.

2.5.5 - Get descriptor

In the project menu, the **Get descriptor** action retrieves the *project descriptor* of this project for later restoration.

2.5.6 - Clean generation data

In the project menu, the *Clean generation data* action commands the ∞Server to clean-up generation data.

2.5.7 - List Builds

Under the status bar are listed all the existing *builds* for the selected project. For each *build*, the page contains:

- The name of the *build* (as defined in the *buildcomment* of the build properties).
- Its unique identifier (for internal use).
- Its creation date.

Several panels are available:

- The *Proxies* panel lists all the servers (∞Server or ∞Proxy) publishing this *build*.
- The *Info* panel lists the tags of the build and its complete description.
- The *Evojump* panel lets the administrator create/download a file image of a packed build.

2.5.8 - Delete a Build

In the *Proxies* panel, it is possible to delete the build from the corresponding ∞Server or ∞Proxies.

2.5.9 - Export a Build

To export a *build* for standalone use, the *build* must first be packed in a secured container - called *evojump*. This is done by clicking on the corresponding button in the *Evojump* panel. A dialog then asks for a passphrase to protect the *evojump*.

Once the *evojump* is packed, two button appears:

- a *delete* button (to remove the corresponding *evojump* from the ∞Server),
- a *download* button (to retrieve the corresponding *evojump* file).

2.5.10 - Delete Projects

In the project menu, the *Delete project* action lets the administrator totally delete a projects.

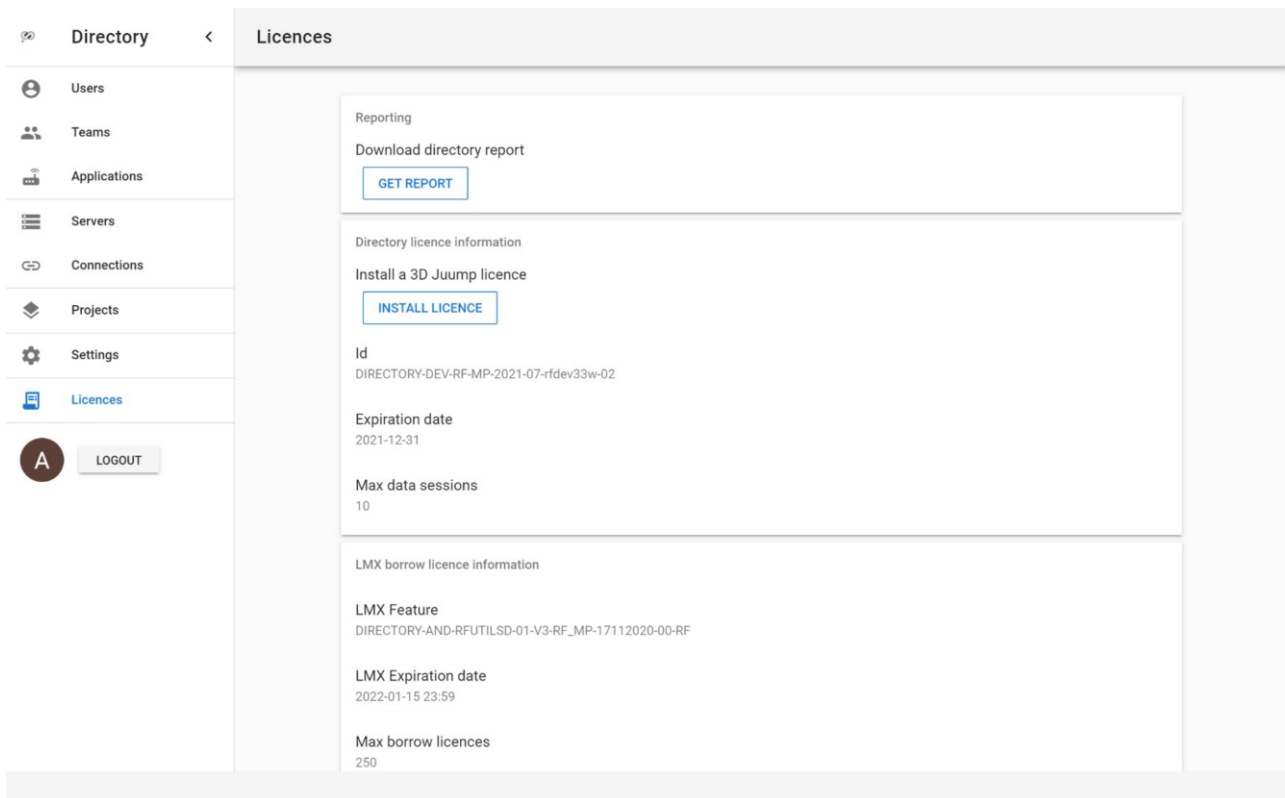
2.6 - Settings

Setting	Range	Description
<i>alloworigin_cache_duration_sec</i>	60 - 86400	allowed origin cache duration in data servers (in sec)
<i>client_hardwarereport</i>	-	enables ∞Client hardware report
<i>client_usagereport</i>	-	enables ∞Client usage report
<i>data_session_max_duration_sec</i>	3600 / 345600	maximum duration of a data session before session revocation
<i>data_session_max_heartbeat_interval_sec</i>	360 / 2700	maximum duration between two heartbeats from ∞Client application before session revocation
<i>data_session_token_lifetime_sec</i>	600 / 14400	data session access token lifetime (in sec)

<code>directory_session_lock_on_ip</code>	-		if true ∞Directory and data sessions will be locked on client ip. It is preconized to register servers by host name when using this security feature
<code>directory_session_max_client_idle_duration_sec</code>	3660 / 345600	/	if there is no user activity during this duration, the ∞Client will close the ∞Directory session
<code>directory_session_max_duration_sec</code>	3660 / 345600	/	maximum duration of the ∞Directory session, if elapsed the ∞Directory session will be closed
<code>directory_session_max_heartbeat_interval_sec</code>	360 / 3600		maximum interval between two ∞Directory session heartbeats, if elapsed the ∞Directory session will be closed
<code>directory_session_token_lifetime_sec</code>	600 / 14400		directory session wait for auth (in sec)
<code>directory_session_wait_for_auth_sec</code>	30 / 900		maximum duration of the authentication procedure, if elapsed the ∞Directory session will be closed
<code>json_web_key_lifetime_hour</code>	2 / 48		lifetime of generated JWK, note that this lifetime should be greater than token lifetime (in hour)
<code>max_dmu_rent_time_sec</code>	86400 / 8035200	/	maximum dmu rent duration
<code>max_token_offline_time_sec</code>	86400 / 8035200	/	maximum duration of lmx borrowed license
<code>proxy_max_reporting_interval_sec</code>	60 / 7200		proxy maximum duration between to directory reporting (in sec)
<code>sso_user_consent_interval_sec</code>	0 / 604800		maximum duration between two user consents. If 0 SSO will be disabled
<code>webhook_report_interval_sec</code>	60 / 604800		duration between two calls to <code>webhook_url</code>
<code>webhook_url</code>	-		a webhook url where ∞Directory and ∞Proxies will POST theirs errors

2.7 - Licences

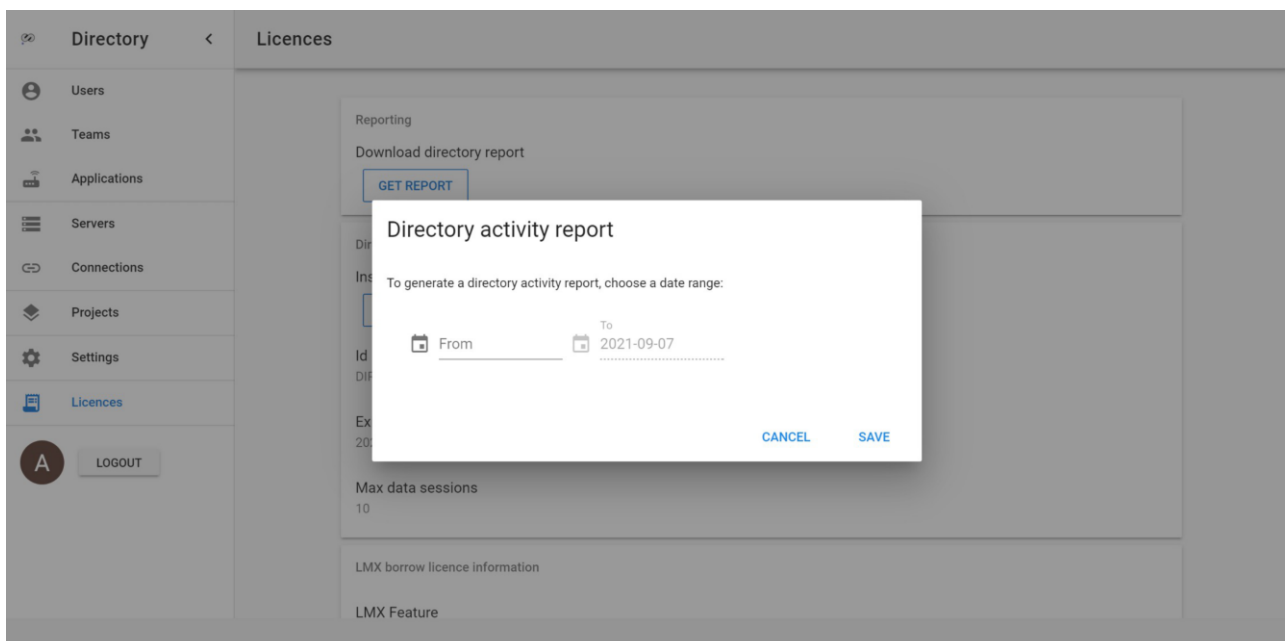
The *License Management* page allows the administrator to : - install a directory license - request the use of tokens for a period of time - list directory licence information - list LMX borrow licence information



Licences

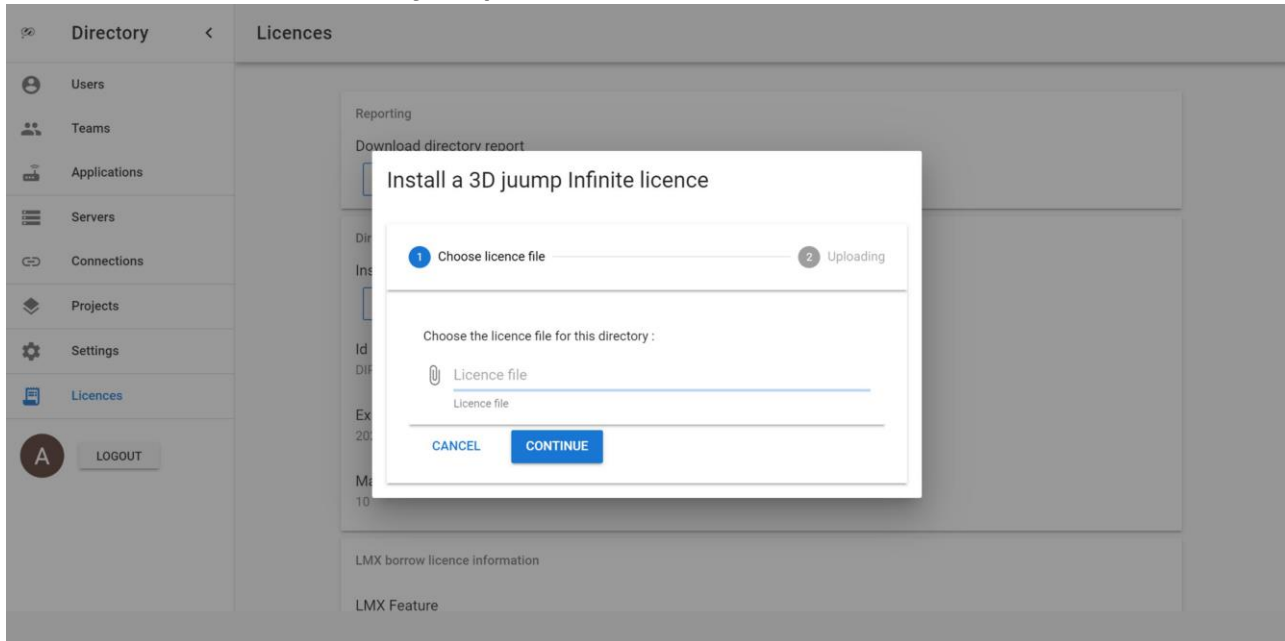
2.7.1 - Activity report

To generate a directory activity report, choose a date range



Activity report

2.7.2 - Install 3D juump infinite licence



Install 3D juump infinite licence

3 - Monitoring

This chapter provides information that will help administrators build automatic tools or services that will report and monitor errors happening on the 3D Juump Infinite servers. It is highly recommended to develop and setup these mechanisms, as they prove very helpful to be immediately reactive and informed when problems occur.

3.1 - Status monitoring

A tool or service may probe the status of the various known 3D Juump Infinite nodes and alert the administrator when a failure is detected.

At anytime, you may probe the status of any 3D Juump Infinite node (∞Directory, ∞Proxy, ∞Server) using, respectively, the directory or proxy API routes `/directory/api/getversion` and `/proxy/api/getversion` with the url parameter `status=true`.

In the case of an ∞Proxy or ∞Server, the returned `status` indicates the status of the node itself. In the case of the ∞Directory, the returned `status` will give an idea of the general status of the 3D Juump Infinite network registered to this directory: - `green`: All the nodes are running correctly. - `yellow`: The Infinite network is partly functional, but some nodes may be down or some services may be unusable. - `red`: The 3D Juump Infinite network is totally unusable.

See the corresponding ∞Directory or ∞Proxy API documentation ("Infinite directory API.html" or "Infinite proxy API.html") for full details.

3.2 - Error notification (Webhook)

A webhook url can be specified in the ∞Directory [Settings](#) to be notified of log entries. Here is a description of the POST content:

```
{
  "numdebug": 0,
  "lastdebug": ["...",...],
  "numinformation": 0,
  "lastinformation": ["...",...],
  "numwarnings": 0,
  "lastwarnings": ["...",...],
  "numcriticals": 0,
  "lastcriticals": ["...",...]
}
```

- The *num** is the number of log entries of this type since last webhook call
- The *last** is a list containing the last three messages of this type

4 - Data access control

3D Juump Infinite **securely** broadcasts DMUs to users. Data protection is deeply rooted into the design of 3D Juump Infinite, all through the [DMU flow](#).

In this paragraph, we summarize the 3D Juump and third-party services required for the 3D Juump back-end.

4.1 - Tcp port usage

3D Juump Infinite brings its own service but also installs third party servers. Make sure your network administrator properly configures these servers in compliance with your network policy.

We summarize which installed third party servers are visible from outside. Tcp port values are the default ones and may vary upon your configuration.

This table present list of TCP port published by ∞Directory, ∞Server, ∞Proxy and ∞StaticProxy.

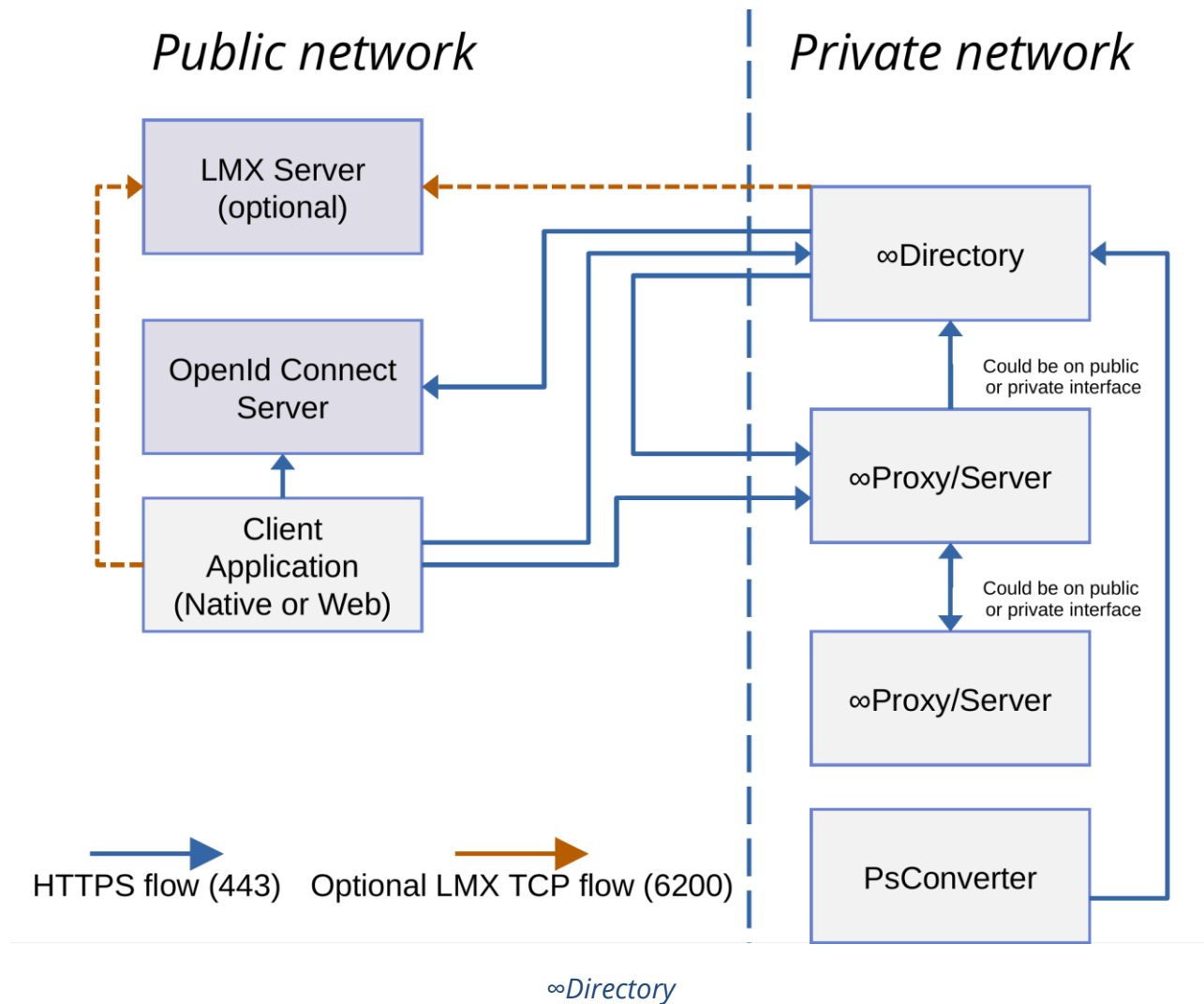
Service	∞Directory	∞Server / ∞Proxy / ∞StaticProxy	LMX Server
Apache (http)	80 (redirected to https)	-	-
Apache (https)	443	443	-
LMX (optional)	-	-	6200

This table present additional TCP port used on loopback by ∞Directory, ∞Server, ∞Proxy and ∞StaticProxy.

Protocol	∞Directory	∞Server / ∞Proxy / ∞StaticProxy	LMX Server
∞Service	2702	2703	-
Apache (https)	443	443	-
PostgreSQL	5440	5440	-
ElasticSearch	-	9200	-

∞Client uses https port 443 to communicate with ∞Directory, ∞Server, ∞Proxy, ∞StaticProxy. In addition to that, the native client may use the LMX Server port 6200 to borrow licenses. The ∞Directory, ∞Server, ∞Proxy and ∞StaticProxy use the https port 443 to communicate with other network elements.

This diagram shows dataflow between components.



4.2 - List of service startup users

This table is only valid for an installation on a Microsoft Windows host. For Linux, the startup user and group are named *juumpinfinite*.

Name of service	Startup user
3dJuump Infinite directory service	Network Service (Service Réseau)
3dJuump Infinite server service	Network Service (Service Réseau)
3dJuump Infinite proxy service	Network Service (Service Réseau)
Apachehttp3djuumpInfinite	Network Service (Service Réseau)
postgresql-x64-13	Network Service (Service Réseau)
elasticsearch-service-x64	Local System Account (Système Local)

The Microsoft Windows startup user is the user able to start the current service. The name in parenthesis is for a French version of Microsoft Windows.

4.3 - Data encryption

In order to ensure the security, all transfers between the nodes of the 3D Juump Infinite network are encrypted using SSL/TLS protocols¹¹. SSL is now deprecated, and apache servers are configured by the install scripts to use only TLS v2 and above protocols. For sake of simplicity we use the acronym *SSL* to designate both SSL and TLS protocols. To establish this encryption, a SSL certificate is mandatory.

4.4 - Data caching policy

The 3D Juump Infinite Client applications can retrieve the *builds* from the ∞Proxy, with the proper authorizations of the ∞Directory. For a secured access to the data, the ∞Proxy cannot be accessed directly and should be registered in a *DMU flow* and always remain in contact with the ∞Directory. The **cached builds deletion** policy has been updated:

- When the ∞Directory cannot join the ∞Proxy, the updated *builds* cannot be broadcasted to the registered 3D Juump Infinite client applications.
- If a ∞Proxy *build* has been deleted through the ∞Directory, it will be deleted by the former until a fresher version is provided by a ∞Server.
- If the link between the source of the data and the given ∞Proxy for a project has been *explicitely* deleted, all the builds belonging to the given project will be deleted by the former.



The ∞Directory denies the *build* replication on a given ∞Proxy? Check for these following points.

- The ∞Directory license is valid?
- Does the ∞Server *project* version match with ∞Directory?
- Is the ∞Server registered and can it be contacted by the ∞Directory? Is the *build* to be replicated actually provided by the ∞Directory?

4.5 - Folder architecture

In case you followed the default folder architecture during the installation, files are organised as follows:

Folder	Usage
<code>\${install_basepath}/current</code>	all files related to the infinite cluster (except binaries) are located there.
<code>\${install_basepath}/current/service</code>	all files related to the ∞Directory, ∞Server, ∞Proxy and ∞StaticProxy.
<code>\${install_basepath}/current/service/3d</code>	all files related to 3d data for the ∞Server, ∞Proxy and ∞StaticProxy.
<code>\${install_basepath}/current/service/logs</code>	logs of the ∞Directory, ∞Server, ∞Proxy and ∞StaticProxy services.
<code>\${install_basepath}/current/postgresql</code>	postgres cluster data and logs.

¹¹ Secure Sockets Layer and its successor Transport Layer Security (TLS) are cryptographic protocols that provide communication security over TCP/IP networks such as the Internet. [wikipedia](https://en.wikipedia.org/wiki/Secure_Sockets_Layer)

<code>\${install_basepath}/current/elasticsearch</code>	all files related to the elasticsearch service for the ∞Server, ∞Proxy and ∞StaticProxy.
<code>\${install_basepath}/current/elasticsearch/data</code>	elasticsearch database.
<code>\${install_basepath}/current/elasticsearch/logs</code>	elasticsearch logs.
<code>\${install_basepath}/current/ssl</code>	private key and ssl certificate files.
<code>\${install_basepath}/current/ssl_pg</code>	on Linux, certificate files used for apache cannot have the same access rights. ssl_pg is a copy if the ssl folder with different access rights.
<code>\${install_basepath}/current/www</code>	all files related to the apache service for the ∞Directory, ∞Server, ∞Proxy and ∞StaticProxy.
<code>\${install_basepath}/current/www/logs</code>	Apache logs.
<code>\${install_basepath}/current/www/webclient</code>	folder containing the web client application for the ∞Directory.
<code>\${install_basepath}/current/www/directory</code>	folder containing the ∞Directory front end web application.
<code>\${install_basepath}/current/www/empty</code>	empty folder used in apache configuration.
<code>\${install_basepath}/current/www/empty/public</code>	public part of empty folder.
<code>\${install_basepath}/current/www/eseditor</code>	folder used to serve eseditor tool.
<code>\${install_basepath}/current/www/eseditor/public</code>	public part of eseditor folder.

4.6 - ∞Server

4.6.1 - DMU Data

The data pushed by a connector into a ∞Server is stored in secured databases that no unauthorized users can access, nor remotely - thanks to the authentication procedure enforced by the Postgres and HTTPS servers, nor locally - thanks to the Operating system authentication procedure.

The data pulled by a ∞Server from a geometry source is stored on disk in a proprietary binary format in a repository that can only be accessed by authorized users. It requires a valid System account with read-access. Remote access is ensured by the HTTPS server for Postgres authorized users only.

4.6.2 - Administration Data

The ∞Server service stores its own credentials (for local server accesses) in a well known cryptographic algorithm with a custom key generation algorithm.

4.7 - ∞Proxy and ∞StaticProxy

4.7.1 - DMU Data

Replicated data is stored in secured databases as specified for ∞Server.

Geometry data is stored on disk in a proprietary binary format, in a repository that can only be accessed by authorized users. It requires a valid System account with read-access. Remote access is ensured by the HTTPS server for Postgres authorized users only.

The ∞Proxy has remote access to parts of the ∞Servers and ∞Proxies databases: its credentials are encrypted and stored in the local secured Postgres database.

4.7.2 - Administration Data

The ∞Proxy service stores its own credentials (for local server accesses) in a well known cryptographic algorithm with a custom key generation algorithm.

4.8 - ∞Directory

4.8.1 - Administration data

Administration data is stored in a secured database that no unauthorized users can access, nor remotely - thanks to the authentication procedure enforced by the Postgres server, nor locally - thanks to the Operating System authentication procedure.

The ∞Directory has remote access to parts of the ∞Servers and ∞Proxies databases: its credentials are encrypted and stored in the same secured database as the administration data.

The ∞Directory service stores its own credentials (for local server accesses) in a well known cryptographic algorithm with a custom key generation algorithm.

4.8.2 - Assets

Assets are stored in secured databases that no unauthorized users can access, nor remotely - thanks to the authentication procedure enforced by the dedicated Apache module, nor locally - thanks to the Operating System authentication procedure.

4.9 - 3D Juump Infinite Client Application

4.9.1 - Connected mode

At start-up, the 3D Juump Infinite client application requires the user to log in to the ∞Directory. No connection with any other node of the 3D Juump Infinite network is initiated unless the ∞Directory has authorized it. Credentials delivered by the ∞Directory for the session are temporary and not reusable.

The 3D Juump Infinite client application itself possesses no credentials to access the ∞Servers, ∞Proxies.

Cached data is stored on disk in encrypted containers.

4.9.2 - Standalone mode

Once a DMU file has been installed to the 3D Juump Infinite DMU Manager or downloaded from a ∞Proxy, its content is made available to any user authenticated to the local computer. 3D data are stored on disk in binary containers (proprietary format, tessellated geometries only).

4.10 - Communications

All network communication between the ∞Directory, ∞Servers, ∞Proxies and 3D Juump Infinite client applications is secured and authenticated by cryptographic protocols:

- DMU-related data and assets are securely transported via TLS.
- Some management informations are securely transported via secured PostgreSQL connection.

4.11 - DMU files

DMU files (*evojump*) are password-protected.

5 - Fault tolerance and preventive maintenance

3D Juump Infinite is designed to ensure availability of the DMU to the connected users. However, like every system, it is subject to fault and breakdown. This chapter gathers a non-exhaustive list of procedures that administrators may choose to apply to keep a high availability of services.

5.1 - Temporary network failure

3D Juump Infinite client application is tolerant to temporary network failure: it does not rely on keep-alive connections.

Temporarily losing network connectivity, will only result in temporary inoperative functions (search, id-card...) and 3D views limited to high-quality 3D data already locally cached. After a short period (minutes), user is forced to quit, see [Settings](#) for more informations.

5.2 - ∞Proxy or ∞Server failure

5.2.1 - Redundancy by ∞Proxy


To circumvent a lasting ∞Proxy or ∞Server failure, one can rely on redundant ∞Proxies.

By design, a ∞Proxy implements a master-slave redundancy solution of its source. Even though the source is dead, connected ∞Proxies will still keep on serving data, effectively maintaining the overall 3D Juump Infinite service. The ∞Directory will route new connection requests to the redundant ∞Proxy and users will thus be able to access and browse the DMU.

5.2.2 - Build backups

To backup a particular *build*, use the *evojump* packing procedure. This *evojump* can later be restored to an ∞StaticProxy or an ∞Server.

5.2.3 - Project backups

To backup a project, just download *project descriptor* from the ∞Directory front end. Use this descriptor to recreate the project on the generator.  Do not recreate the project on several ∞Server, a project should be available on only ONE ∞Server.

5.2.4 - ∞Server backups

To backup data from a ∞Server, backup [Builds](#) and [Projects](#)

5.2.5 - Asset databases

The assets databases are hosted on a regular postgres server on the ∞Directory and thus supports classical Postgres preventive maintenance procedures. Particularly, it is good practice

to maintain a secondary postgres server and initiate a continuous master-slave replication or scheduled punctual replications for future restoration.

5.3 - ∞Directory failure

5.3.1 - Administration web interface

The HTTP server in the ∞Directory only serves administration pages which are not critical for the DMU broadcasting service. Though the lasting unavailability of the administration front-end is of course an issue, it usually does not bear the same level of criticality and may thus not require specific preventive maintenance procedures.

5.3.2 - Authentication service

The ∞Directory authentication service is the keystone of 3D Juump Infinite and its lasting failure **will** imply a downtime. However, one can seek to limit the downtime period.

To maintain the authentication service running, the ∞Directory needs four components:

- its HTTP server,
- its Postgres database,
- its ∞Directory service,
- the LM-X server.

Thus, it is preferable to schedule and store regular backups of the Postgres database on an external drive:

1. Backup:

- Execute the following command-line:

```
pg_dump.exe --encoding=UTF-8 --format=custom --exclude-schema=public --host={HOST}
--port={PORT} --username=postgres --password --dbname=InfiniteDirectory --file=
directory_dump_custom.pg
```

2. Restore:

- Reinstall the ∞Directory if needed.
- Stop the ∞Directory service.
- Using a SQL console (or pgAdmin), execute the following query:

```
DROP SCHEMA IF EXISTS "Directory" CASCADE;
DROP SCHEMA IF EXISTS "DirectoryLogs" CASCADE;
```

- Execute the following command-line:

```
pg_restore.exe --exit-on-error --format=custom --host={HOST} --port={PORT} --user
name=postgres --password --dbname=InfiniteDirectory directory_dump_custom.pg
```

- Restart the ∞Directory service.

Having a spare ∞Directory is also good practice. Should the first one fail, it would only require a Postgres database restoration to put a new ∞Directory up and running.

5.4 - LM-X failure

3D Juump Infinite relies on LM-X software protection system. It is mandatory that your LM-X system is properly set and maintained. Please carefully read the LM-X documentation for information regarding maintenance and best practice.

6 - Migration procedures

6.1 - Cluster migration

To migrate a cluster:

1. Export the descriptors of all projects (from the ∞Directory)
2. Pack and download any useful build from all projects (from the ∞Directory)
3. Delete all projects from the cluster
4. Stop all servers (∞Directory, ∞Servers and ∞Proxies)
5. Install the new ∞Directory software, then restart the ∞Directory server
6. Install the new ∞Server software, then restart all ∞Server servers
7. Install the new ∞Proxy software, then restart all ∞Proxy servers
8. Recreate all projects from their descriptors
9. Migrate the builds (see the corresponding chapter in the Integration Manual)
10. Upload and unpack the migrated builds

6.2 - Assets migration



Migrating from 3.0 and below requires that all assets are patched to reflect the new authentication mechanisms. The assets' security descriptors have to be updated to reflect the new team and user identifiers.

1. Before starting the procedure, make sure all asset databases are properly backed up
2. Retrieve the internal team identifiers from the old cluster
3. Migrate the cluster (see above)
4. Recreate the teams
5. Prepare a map of all users, matching the previous user identifier to the corresponding subid from the new *Authenticator*
6. Prepare a map of all teams, matching the previous team identifier to the corresponding team identifier from the new ∞Directory
7. Patch all assets, replacing previous user and team identifiers in the security descriptor.



In versions of 3D Juump Infinite 3.2 and below, assets were stored in a CouchDB database. Assets are stored in the ∞Directory postgres database starting with 3D Juump Infinite 3.3. To migrate assets from a 3.2 version to a 3.3 version, we advise to follow these steps:

1. Before starting the procedure, make sure all asset databases are properly backed up.
2. Download all the assets (couchdb documents) from the couchdb database while your previous ∞Directory 3.2 is running.

3. Within the security descriptor of each asset, replace 3.2 team and user identifiers by their 3.3 counterparts. Make sure that no old identifier remains, or it will result in inaccessible assets.
4. Upload the patched assets to the ∞Directory 3.3, using the ∞Directory asset API (in particular, the [directory/api/assets/{projectid}/docs](#) endpoint). See the corresponding ∞Directory API documentation ("Infinite directory API.html") for full details.

Native client

The ∞Client is composed of several software components:

- the *Hub* is a central element responsible for user management, token management, add-on management, etc.
- the *Browser* is a mandatory add-on that is launched from the *Hub* and lets the (authenticated) user visualize and interact with the chosen DMU.
- the *Local DMU Manager* is an optional add-on that adds the ability to install data locally in *offline* and *standalone* modes (see below).

1 - Modes

The ∞Client supports two distinct modes:

- In *online* mode, the ∞Client is connected to a remote ∞Directory and browses an *online DMU*, i.e. data streaming from the 3D Juump Infinite cluster.
- In *standalone* mode, the ∞Client browses *standalone DMU*, i.e. local data installed from *evojuump* files.

1.1.1 - Online mode

The *online* mode requires:

- that the user can physically contact a remote ∞Directory,
- that the user has valid credentials for this ∞Directory,
- that the credentials give access to at least one *online DMU*,
- that a *floating licence token* is available or a non-expired *node-locked license*.

Once the user is properly authenticated and once a DMU has been selected, the *Hub* launches the *Browser*. A *floating license token* is locked until the *Browser* is closed. Should the local computer loses its *floating license token* while the *Browser* is running (due to a network error, for instance), the *Browser* will forcibly close two minutes after issuing a warning.

1.1.2 - Standalone mode

The *standalone* mode requires:

- that the *Local DMU Manager* add-on is installed and running on the local computer,
- that a valid non-expired *node-locked license* is installed on the local computer or non-expired_borrowed token,
- that the user has installed a *standalone DMU*,
 - by importing a valid file to the *Hub*,
 - by providing the password corresponding to the file.

Once a *standalone DMU* has been selected, the *Hub* launches the *Browser*. Should the *node-locked license* expire while the *Browser* is running, the *Browser* will forcibly close two minutes after issuing a warning.

2 - Host requirement

2.1 - Hardware

The client application can run on a broad range of CPUs, from a mobile to a high-end x86-64 central processing unit (CPU). The amount of CPU power and RAM required depends on the complexity of your DMU and the availability of a dedicated graphic processing unit (GPU).

The dedicated graphic card (GPU) is not required but when it is available, it should be installed with an OpenGL 3.1 and above driver. Any NVIDIA GeForce 8 and above, AMD Radeon 4xxx and above or Intel HD4000 and above may fit. Without a dedicated GPU, the 3D Juump client application is able to run with the integrated GPU.

Eventually, you should also reserve enough hard disk space to store for your digital mock-up data and workspaces. The ∞Client application and Local DMU Manager installed size is only a negligible fraction of the occupied size, respectively 200MB and 800MB.

	CPU	RAM	GPU
∞Client minimum	2 cores	2GB	OpenGL 3.1 1GB
∞Client recommended	4 cores	4GB	OpenGL 3.1 2GB
Local DMU Manager	+ 2 cores	+ 2GB	

2.2 - Software

Supported Operating System: **64bit Microsoft Windows 7 and above.**



In order to install the Local DMU Manager on the user host, you must have an **administrator** account. Note the *administrator* account is only required for the installation. Once the application is deployed, the user can have access to their data with its regular account.

3 - Installation

3.1 - Installing the *Hub* and *Browser* add-on

Note: the *Browser* add-on is automatically installed with the *Hub*.

3.1.1 - Requirements

∞Client minimal requirements:

- Windows 7/8/10 64-bit version
- Dual Core CPU x86-64
- 4GB RAM
- 4GB disk space for binaries and caches
- GPU with OpenGL 3.1 and GLSL 140 support with 3GB RAM

∞Client recommended requirements:

- Windows 10 64-bit version
- Quad Core CPU x86-64
- 6GB RAM
- 8GB disk space for binaries and caches (SSD)
- GPU with OpenGL 3.1 and GLSL 140 support with 4GB dedicated RAM

Keyboard and mouse are mandatory for a full use of 3D Juump Infinite. Touchscreen support is limited to navigation.

3.1.2 - Installation procedure

To install the 3D Juump Infinite *Hub* and *Browser*:

- Double click on the *3DJuump Infinite X64-setup-admin-2.x.x.xxx.exe* installer (requires administrator privileges).
- Validate the license agreement.
- If required, change the installation folder.
- Then, if needed, modify the program folder in the start menu and choose if you want to create a shortcut on the desktop.
- Click on the *Install* button to start the installation.

The installer will deploy the *Hub* application and its mandatory *Browser* add-on, plus any dependencies (including *Visual C++ Redistributable*).

Note: an alternative installer is also available for standard users lacking administrator privileges. This installer does not deploy any `_Redistributable`, hence it is only usable on computers already equipped with the following:

- Visual C++ Redistributable 2015 x64
- Visual C++ Redistributable 2008 x64 SP1 ATL

3.1.3 - Default settings configuration

3.1.3.1 - Location

The settings of the *Browser* application are stored in `settings.ini` files found in the following locations:

- `%PROGRAMDATA%\3DJuumpInfiniteX64\settings.ini` (global settings for all users)
- `%APPDATA%\3DJuumpInfiniteX64\settings.ini` (current user's settings)

3.1.3.2 - Resolution order

For a given settings key, the value is read from the global settings location first and then, if the key is not present, it is loaded from the current user's settings. It means that the administrator is allowed to force a settings for all users. For example, the administrator can ship a predefined global `settings.ini` file with his installation program.

3.1.3.3 - Useful settings keys

Several settings can (and probably *should*) be set by the administrator:

- `Security/verifySSLPeer`: Tells 3D Juump Infinite whether it should verify the SSL certificates of the `∞Directory` and `∞Server/∞Proxy` it connects to. Turn it off when using internal self-signed certificates.
- `Security/enableHttpProxy`: Tells 3D Juump Infinite whether it should use the system-defined HTTP proxy for communicating with the 3D Juump Infinite cluster.
- `HUB/allowLocalAccount`: Tells the *Hub* whether it should display the local account. Turn it off if you don't want your users to ever work in standalone mode.

3.1.3.4 - Pre-registering `∞Directory`

`Directory` could be pre-registered per user by adding an entry in the `%APPDATA%\3DJuumpInfiniteX64\directories_3_3.json` file, or globally by adding an entry in the `%PROGRAMDATA%\3DJuumpInfiniteX64\directories_3_3.json` file. Global entries could not be removed from the *Hub*.

```
[
  {
    "directoryName": "MyDirectory",
    "directoryUrl": "https://hostname/directory",
    "licenseServer": "",
    "serverLicenseFeature": ""
  }
]
```


3.2 - Installing the *Local DMU Manager* add-on

3.2.1 - Requirements

To run the *Local DMU Manager*, the local computer must meet some minimal requirements:

- Windows 7/8/10 64-bit version
- 4GB RAM
- 20GB disk space for binaries and local DMUs

Note: depending on the size and number of your local DMUs, the *Local DMU Manager* may require more disk space.

Note: the *Local DMU Manager* will take advantage of an SSD for store the local DMUs

3.2.2 - Installation procedure

To install the 3D Juump Infinite *Local DMU Manager*:

- Double click on the *3DJuump Local DMU Manager X64-setup-2.x.x.xxx.exe* installer (requires administrator privileges). Please note that versions of the Local DMU Manager and the Hub must be the same.
- Validate the license agreement.
- If required, change the installation folder.
- Then, if needed, modify the program folder in the start menu and choose if you want to create a shortcut on the desktop.
- Click on the *Install* button to start the installation.

The installer will deploy the *Local DMU Manager* add-on plus any dependencies, including the following *Visual C++ Redistributable*:

- Visual C++ Redistributable 2015 x64
- Visual C++ Redistributable 2010 x86
- Visual C++ Redistributable 2008 x64 SP1 ATL

3.2.3 - Local web application

It is possible to run single page web application in *Standalone mode*. Edit *%PROGRAMDATA%\3DJuumpLocalDMUManagerX64\Settings.ini* add following entry *LocalWebApps=path_to_web_app_folder* in *[localserver]* section. Web application folder should contain a subfolder per web application, each subfolder should contain an *index.html*.

3.2.4 - ElasticSearch configuration

Depending on computer available RAM and size of DMU it might be needed to adjust amount of memory dedicated to ElasticSearch instance. This is achieved by editing *%AppData%\3DJuumpLocalDMUManagerX64\databases_3.3/elasticsearch/config/jvm.options*. See [ElasticSearch documentation](#) for details.

Third party application protection

When deploying a webservice based on the Infinite web API you may want to add a security layer to limit access to your web application and data. This can be done by using an ∞Directory session access token or an ∞Proxy data session access token.

1 - Defining user access rights

Add tags to users or teams that will be allowed to access your web application. [See Third party application user access rights](#)

For example, you can define two scopes destined to allow respectively read and write operations:

```
thirdpartyscopes:myapplication.read thirdpartyscopes:myapplication.write
```

2 - Request scope

When opening an ∞Directory session from the web API, add the desired application claims.

```
let lDirectorySession : DirectorySessionInterface = ...;  
lDirectorySession.getPopupBasedAuthenticationUrl(..., ['myapplication.read', 'myapplication.write']);
```

3 - Protect your web application

Use the Directory session access token returned by

```
let lDirectorySession : DirectorySessionInterface = ...;
lDirectorySession.getAuthenticationBearer();
```

or the Data session access token returned by

```
let lDataSession : DataSessionInterface = ...;
lDataSession.getDataSessionBearer();
```

in HTTPS calls to your backend (with the header *Authorization: Bearer <token>* or as a query parameter).

4 - Validate token and claims

First, validate the received token using */directory/api/.well-known/jwks.json* or *directory/api/instrospect* endpoints. Then ensure that the *scope* field contains the expected claims.

Directory session access tokens will only contain application scopes. It allows you to ensure that the user is logged on the ∞Directory and has the expected claims. In addition, Data session access tokens contain extra information related to the Project and Build, that may be used to limit access based on project access rights.

Example of a Directory session access token

```
{
  "access:client": true,
  "exp": 1609866581,
  "iat": 1609866282,
  "ip": "...",
  "iss": "7fed45f6326f41fab39f27f8a7271702",
  "native": false,
  "nbf": 1609866282,
  "oidcsub": "...",
  "salt": "7b4pzw",
  "scope": "myapplication.read myapplication.write",
  "session_eol": 1609952681,
  "sessionId": "1ab88920ebc14f3baee8ac4eb5aa301e",
  "tokentype": "TT_Directory"
}
```

Example of a Data session access token

```
{
  "access:build": true,
  "buildid": "{d45269d4-9be1-4ff3-b02f-e9ca60ab7144}",
  "buildtags": [
    "MyBuildTag"
  ],
  "exp": 1609868083,
  "iat": 1609866283,
```

```
"ip": "...",
"iss": "7fed45f6326f41fab39f27f8a7271702:this",
"nbf": 1609866283,
"pglogin": "...",
"pgpwd": "...",
"projectid": "prj_9c9a52a4e220395c93458a0d5e7b3f79",
"salt": "RQrr9w",
"scope": "myapplication.read myapplication.write {d45269d4-9be1-4ff3
-b02f-e9ca60ab7144} prj_9c9a52a4e220395c93458a0d5e7b3f79",
"sessionid": "{c1dcad46-344a-4465-84d9-8539c7c29e2b}",
"srcmodelpoolver": "1~c966f5574e2e44d3b50d0677f6ba1cd7",
"tokentype": "TT_DataSession"
}
```

Annexes

These annexes contain:

- the release notes,
- the security guidelines,
- the list of third-party licenses,
- the list of known limitations.

1 - Release notes

1.1 - 02/16/2025: version 3.3.16

Backend

- Fixed an incorrect behaviour of retry options ('retrymaxtimeperworkersec' and 'retryworkercount') of the PsConverter when they were set to 0

Native Client

- Fixed a bug where multi-layer metadata exports produced no output

1.2 - 18/02/2024: version 3.3.15**Hub**

- Fixed license file installation with non ascii character in path

Backend

- Increased max instance modifiers from 1024 to 65536

Native Client

- Fixed Post exporters not working

1.3 - 26/01/2024: version 3.3.14**Directory**

- Relaxed constraint on query parameters check on OpenId Connect redirect url.

1.4 - 16/01/2024: version 3.3.13**Native Client**

- Fixed LTA banner display

Hub

- Added a way to configure elasticsearch memory for Local DMU Manager

1.5 - 14/12/2023: version 3.3.12**PsConverter**

- Fixed a bug causing instance metadata to be discarded (fix from 3.3.11 was incomplete)

1.6 - 8/12/2023: version 3.3.11**Hub**

- Added LTA license support

Native Client

- Fixed url encoding of attached documents file path

WebAPI

- Fixed various bugs

PsConverter

- Added a new *subpartLevel* option
- Fixed a bug causing instance metadata to be discarded

Backend

- Added a userinfo api endpoint

1.7 - 10/10/2022: version 3.3.10

Native Client

- Fixed slow 3d/bom export
- Added support of a global directories_3_3.json configuration file

Generator

- Added product structure filtering per configuration during generation

PsConverter

- Added optional file prefix for generated rub files

Web Api

- Fixed cache issue causing bad visibility or coloring
- Introduced many major changes and functionalities. Please refer to the javascript 3djump web api for more details.

1.8 - 23/05/2022: version 3.3.9

PsConverter

- Added a new field *psconverter:badxform*
- Added an option to remove instantiation and bake xforms

Backend

- Updated PostgreSQL to version 13.7
- Updated LibPq to version 13.7
- Updated Apache to version 2.4.53

1.9 - 09/05/2022: version 3.3.8

Backend

- Updated Fbx sdk to version 2020.3.1
- Added Loki/Grafana log handler

Native Client

- Improved hub error log message

1.10 - 02/05/2022: version 3.3.7

Backend

- Fixed minor issues

Installer

- Fixed native client redirect url registration

Web API

- Added npm package

Web Client

- Fixed minor issues

1.11 - 01/04/2022: version 3.3.6

Administration Interface

- Added an administrator icon in users view

Backend

- Fixed build error message not cleared

PsConverter

- Regrouped namespaced metadata (eg: 0000123456...0000123456-1::lxyG)

Web Client

- Added script injection protection

Native Client

- Improved callout editor

1.12 - 15/03/2022: version 3.3.5

Native Client

- Fixed upload task issue when using Azure user id

Backend

- Improved PlmXml support

1.13 - 10/03/2022: version 3.3.4

Local DMU Manager

- Added local web application support

Administration Interface

- Fixed minor issues

Backend

- Fixed minor issues

PsConverter

- Added options to retry out of memory jobs with less workers and bigger memory budget

Web API

- Fixed minor issues

Web Client

- Fixed minor issues

1.14 - 26/02/2022: version 3.3.3**Native Client**

- Fixed minor issues

Web Client

- Fixed minor issues

Administration Interface

- Fixed minor issues
- Improved build quality indicator
- Disabled new users to avoid access to untagged applications that can give access to data

Backend

- Added a way to substitute OpenId Connect user unique id
- Updated PostgreSQL to version 13.6

Install

- Added a home page

1.15 - 26/01/2022: version 3.3.2**Native Client**

- Added copy annotation svg to clipboard
- Fixed copy annotation text to clipboard
- Fixed keyword type search and filter
- Fixed minor issues

Web Client

- Fixed minor issues

Administration Interface

- Added a way to download build generation log
- Added a build quality indicator

Backend

- Added an api /generationreport and /generatebuildresult to retrieve generation build result and logs of a build
- Fixed minor issues in doc store api
- Fixed error when uploading several files through /pushfile api
- Fixed rare dead lock of the proxy while restoring a build

Third Party

- Updated Apache to version 2.4.52 (Windows)
- Updated Elasticsearch to version 6.8.23 with log4j 2.17.1 (Windows)
- Updated liburiparser to version 0.9.6 (Windows)

1.16 - 03/01/2022: version 3.3.1

Native Client

- Fixed a bug in annotation task

Backend

- Updated PostgreSQL to version 13.5
- Updated Apache to version 2.4.51
- Updated Elasticsearch to version 6.8.22 (fix log4j known vulnerabilities)
- Added an api /pushfile to upload source geometry files

Install

- Added an option to install even if OpenId Connect server is not reachable

1.17 - 01/12/2021: version 3.3.0

Native Client

- New annotation renderer
- Improved search behavior

Backend

- Removed Couchdb
- Added alternative urls
- Improved performances of HTTP API
- Changed annotation model to improve support
- Allow use of OpenId Connect access token
- Remove PostgreSQL communication between backend nodes

Web API

- Added annotation support

Web Client

- New web client design
- Added annotation support
- Added multi cut plane
- Added multi filtering layer
- Added coloring layer
- Added undo/redo functionality

2 - Security guidelines

3D Juump Infinite relies on several third-party server-side software components that, like any software, are subject to security vulnerabilities. Make sure you apply the latest security patches on the software components used.

List of software components used in version 3D juump infinite 3.3

- PostgreSQL version 13.x
- Elasticsearch 6.x
- Apache Lounge Web-server 2.4.53 : though any version sharing the same major version should be compatible.
- OpenSSL : the version of openssl may change depending on the system used. versions 1.0.2 1.1.0 and 1.1.1 are supported.

3 - Third-party software licenses

The details of licenses is available on the 3D Juump Infinite Third Party License manual provided with the software.

4 - Range of use

This annex summarizes the range of use of the software.

4.1 - Minimum requirements

The ∞Directory, ∞Server, ∞Proxy and ∞StaticProxy software run on any of the following operating systems with an **IPv6 stack**:

- Microsoft Windows 7 and above, 64-bit version
- Linux Debian 11 bullseye, AMD64
- Linux Ubuntu 20.04 LTS (focal) AMD64

Minimum hardware requirements are:

- Quad-core processor (support of POPCNT x86 instruction is mandatory, support of CRC32 x86 instruction is recommended)
- For the ∞Directory, ∞Proxy and ∞StaticProxy : 8GB of RAM
- For the ∞Server : 4GB of RAM by CPU Core (A Quad-core processor should have at least 16 GB of RAM).
- 1GB disk space for binaries + sufficient disk space for data (depending on your data sources)
- High-speed hard drive disk or solid-state drive highly recommended

∞Client minimal requirements:

- Windows 7/8/10 64-bit version
- Dual Core CPU x86-64
- 4GB RAM

- 4GB disk space for binaries and caches
- GPU with OpenGL 3.1 and GLSL 140 support with 3GB RAM

∞Client recommended requirements:

- Windows 10 64-bit version
- Quad Core CPU x86-64
- 6GB RAM
- 8GB disk space for binaries and caches (SSD)
- GPU with OpenGL 3.1 and GLSL 140 support with 4GB dedicated RAM

Keyboard and mouse are mandatory for a full use of 3D Juump Infinite. Touchscreen support is limited to navigation.

4.2 - Supported input formats

3D Juump Infinite is able to process the following formats:

- 3D Experience (.3Dxml) - 2014 => 2015x
- 3DM OpenNurbs – Rhino (.3dm) (Windows only)
- 3DS (.3ds)
- ACIS (.sat) - all => R21
- ASC Medusa 3D (.asc)
- CADDs (explicit parts) & CAMU (. _pd, . _ps) - 4 & 5
- CATIA V4 (.model,.dlv,.exp,.session) - all 4.xx
- CATIA V5 (.CATPART,.cgr,.CATProduct) - R10 - R26
- CATIA V6 (.3Dxml) - 2011x => 2013x
- FBX (.fbx)
- glTF v2 (.gltf, .glb)
- I-DEAS (.arc,.unv) - all => NX5
- IGES (.igs) 5.2 & 5.3
- Inventor (.ipt,.iam) - all => 2017
- JT-Format (JtOpen) (.jt) 7.0 => 10.2
- Matra Euclid 3 (.e3i) - 3.2
- Nastran (.nastran)
- NX Unigraphics (.prt) - 11 => NX11
- OBJ (.obj, .mtl)
- Parasolid XT-Format (.x_t) - all => 28
- PlmXml (.plmxml) - schema v4 *experimental*
- ProEngineer (.asc,.prt, .neu) - part files: 13 => Creo 4 (F000) / neutral files: 13 => WF5
- Rhino 3D (.3dm)
- ROBCAD (.rf)
- Solidworks (.sldprt,.sldasm) - 99 => 2017
- STEP AP203, AP214 and AP242 (.stp) - 203/214/242
- Straessle EUKLID (.edx)

- STL (.stl)
- VDA (.vda)
- VRML (.wrl, .wrz, .vrm) - 97

4.2.1 - Geometry

3D Juump Infinite only accepts surface information. All other geometric informations are ignored (in particular, vector and point data are not supported).

4.2.2 - Metadata

3D Juump Infinite is able to extract product structure, including external references. It also extracts textual, numeric and datum metadata, plus any combination of lists and maps of the above.

4.2.3 - Annotation

3D Juump Infinite is able to retrieve both FTA and PMI. It is limited in the number of options it supports regarding the FTA and PMI representation. Supported data includes:

- text content,
- fonts,
- colors,
- shapes (limited to square and flags),

Symbols (NOA) are not automatically read from input files and require a manual customization to circumvent the lack of universal symbol catalog.

All glyphs visible in a given Annotation View must fit in a 2048x2048 pixels image.

4.3 - Supported output formats

4.3.1 - Geometry

Supported output formats for geometry export are:

- FBX
- GLTF2
- JT
- OBJ
- STEP¹²+JT
- STEP+WRL
- VRML
- WRL+WRL
- WRZ+WRZ

¹² STEP AP242 Part 21

Exported geometries are tessellated. Annotations are not exported.

4.3.2 - Image

Supported output formats for image export are:

- JPEG
- PNG
- TIFF

Transparency support depends on the output format.

4.3.3 - Metadata

Supported output formats for metadata export are:

- CSV (comma separated)
- CSV (semi-colon separated)
- JSON
- XML

4.3.4 - Presentation

Supported output formats for presentation export are:

- HTML
- HTML with DZSlide embedded viewer
- JSON
- Markdown
- ODP

Note: Markdown and ODP do not support rich-text, thus when slide comments containing rich-text are found, they are exported as raw-text instead.

4.4 - Limits

Known limits include:

- Minimum number of assemblies¹³: 1
- Minimum number of single parts¹⁴: 1
- Maximum length of utf-8 document ids (in byte) : 255
- Maximum number of instance modifiers per structure document: 1024
- Maximum number of parts¹⁵: 268 435 455

¹³ Structure document with children

¹⁴ Structure document with geometric representation

¹⁵ PartMetadata documents

- Maximum number of links¹⁶: 268 435 455
- Maximum number of instance metadata¹⁷: 536 870 911
- Maximum number of ids¹⁸: 1 073 741 823
- Maximum number of geometry instances¹⁹: 16 777 215
- Maximum number of distinct materials: 32 767
- Maximum number of annotation views: 1 073 741 823
- Maximum number of annotation types: 1 024
- Maximum number of annotations visible at the same time: 262 143
- Maximum number of conf declared by the connector: 10 000
- Maximum number of instance modifiers per structure document: 65 535
- Maximum number of *nested* sub objects per metadata document: 10 000
- Maximum number of *configured* field in index mapping: 8
- Maximum number of bytes for *text* and *keyword* metadata fields: 32 767
- Maximum scene size : [-2e6;+2e6] around origin along each axis
- Maximum number of clauses per search/filter : 10 000
- Maximum number of terms per attribute filter : 65 536
- Maximum number of attribute values to allow enumeration : 1024
- Maximum size of attribute value to allow enumeration (in byte) : 100

5 - Export control classification

The Software, which integrates dual use information security items of American origin (ECCN 5D992.c <10%), is subject to the US Export Administrative Regulations (EAR) 15 C.F.R. part 730 et seq. for the country Group E:1 and E:2 which are, at the date of the License Terms and Conditions: Iran, North Korea, Sudan, Syria and Cuba. In particular, the User shall not use, export or re-export the Software in those countries and with end users or for end uses in breach of the US export control regulations.

The Software has been the subject of a declaration of operations relating to a means of cryptology to the ANSSI (Declaration N ° 17070363). However, the Software does not come under Regulation (EC) N ° 428/2009 of May 5, 2009, setting up a Community regime for the control of exports, transfer, brokering and transit of dual-use items, as confirmed by the Direction Générales des Entreprises / Services des Biens à Double-Usage Goods in his mail N ° FR 80404.

¹⁶ LinkMetadata documents

¹⁷ InstanceMetadata documents

¹⁸ Distinct *id* fields

¹⁹ One source model instantiated at one world position

