

Team For Capella

Installation guide

Version v7.1.1, 2026-11-17

- Scope
  - Identification
    - Description of the product
    - Deployment modes
    - Applicability
  - Document overview
- Referenced documentation
- Team for Capella Server installation
  - Installation Architecture
  - Recommended system requirements
    - Server computer recommended system requirements
    - Client computer recommended system requirements (only for local client installation)
  - Deployment recommendations
    - Network
    - Scalability and size of models
    - Disclaimer
  - Installation procedure
    - Team for Capella Server installation procedure
      - Configure the default WSS/HTTPS connection
    - Extensions installation
    - Jenkins scheduler installation
      - [installation-guide:jenkins_offline_installation_procedure:::]
    - Administration features installation
    - License software installation
    - Installation verification
  - Team for Capella server configuration
  - How to reuse a previous server configuration with a v7.1.0 server
    - Uninstallation procedure
- Team for Capella Client installation
  - Requirements
    - Recommended system requirements
  - Installation Procedure
    - Team for Capella Client installation procedure
      - Default client installation
      - Scripted client installation

            - Manual client installation
            - Extensions installation
        - Team for Capella Client configuration
            - Multi-user client installation
        - Verification installation procedure
- Team for Capella license software installation
- Migration process
- Version compatibility
    - Model migration from previous version to v7.1.0
    - Capella Addon migration
    - Model migration from an older version
        - Changes for Server Installation from v7.0.1_202607 to v7.1.0
        - Changes for Server Installation from v7.0.1_202509 to v7.0.1_202607
        - Changes for Server Installation from v7.0.1 to v7.0.1_202509
        - Changes for Server Installation from v7.0.0 to v7.0.1
        - Changes for Server Installation from v6.1 to v7.0.0
        - Changes in Server Configuration
        - Changes in Jenkins Jobs Installation/Configuration
        - Changes in Jenkins Jobs
        - Other Changes
- Performance consideration feedback
    - Reference time
        - Case Studies
        - Measures
            - Computer used for the tests
            - Reference elapsed time
        - Analysis and conclusion
    - Dedicated tests on Combined IFEs
        - Server
        - Client
        - conclusion
    - How to get characteristics of your model

Chapter 1. Scope

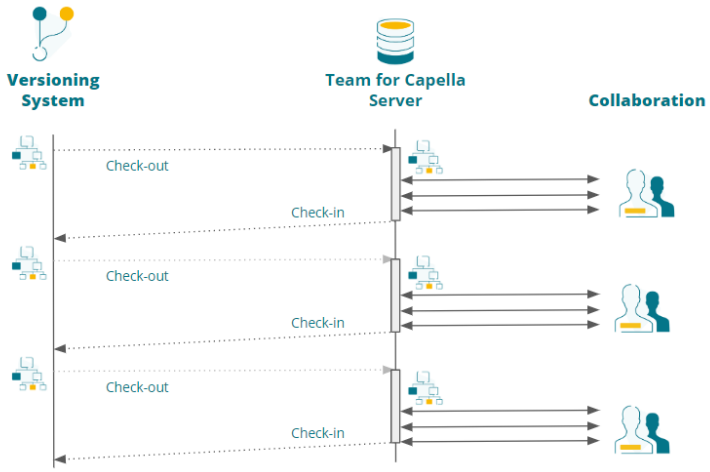
Identification

Description of the product

Team for Capella is a collaborative solution to have several contributors working on the same model, with the granularity as fine as a model element and diagrams.

It decouples the versioning issue which is ensured by a Source Control Management (SCM) tool from the concurrent accesses issue.

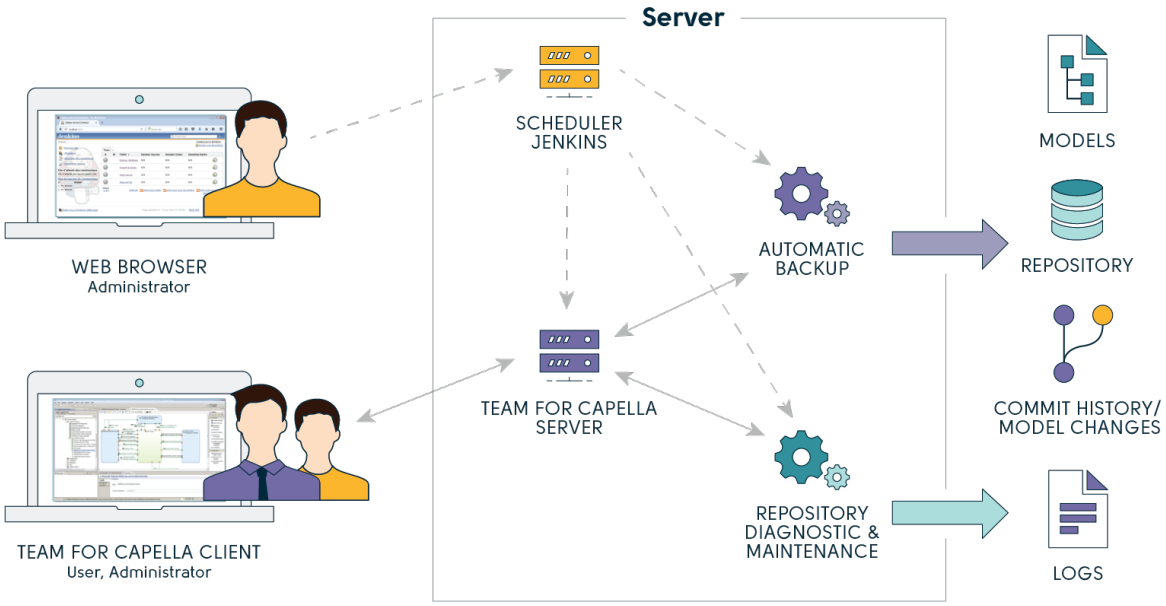
It introduces a shared repository which is populated from the SCM tool and which enables several users to work on the same model.



Users can simultaneously edit the same Capella model without conflicts. Only modified elements are locked, not the whole model, and other users can see in live modifications made by connected teammates. It is no longer necessary to split the model into fragments or to merge concurrent modifications.

It is recommended to version exported models with a SCM tool, for instance Git. Please refer to the Capella embedded documentation at the section Capella Guide > User Manual > Version Control with GIT for more details.

Here is an overview of the Team for Capella architecture:



Team for Capella is composed of three parts:

- A server, to manage the model repositories, and associated features (such as locks, etc.);
- An administration module, to schedule automatic backups (models, changes and database) and trigger diagnostic and maintenance tasks;
- An add-on, packaged as an update site, to bring the multi-user functionalities on top of the standard Capella rich client.

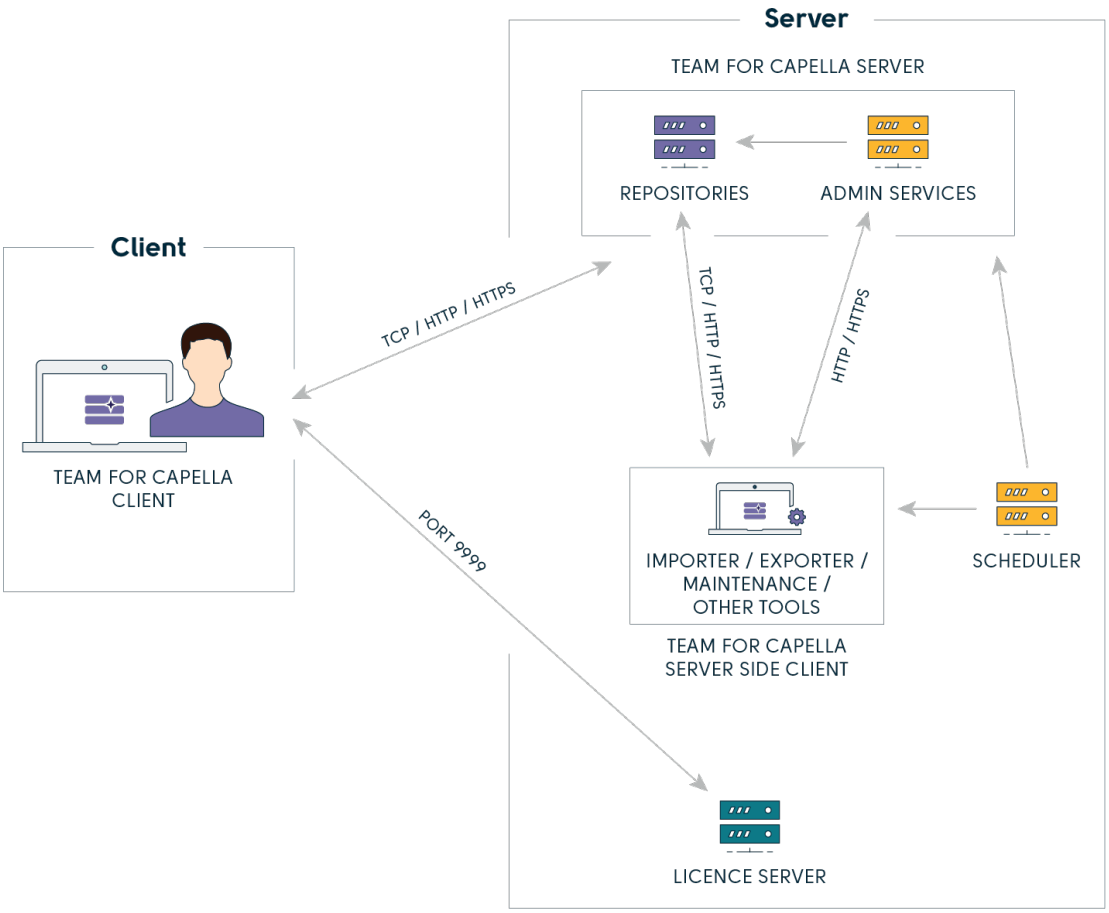
Please refer to the Team for Capella documentation at the section Introduction to Team for Capella > Rationale and Concepts for more information.

Team for Capella is available in 64 bits version for Windows and for Linux.

The Team For Capella server is composed of the CDO repositories server and an embedded Jetty server.
By default, the Jetty admin server is automatically started with the CDO server in HTTPS mode on port 8443. The same secure endpoint carries WSS repository traffic.

The REST admin server is used:

- to manage repositories with the REST Admin API
- by applications (importer, maintenance application, console application) to execute code on server



Please refer to the Team for Capella documentation at the section System Administrator Guide > Server Configuration for more information.

Deployment modes

Team for Capella can be deployed on different modes:

- **Cloud:** With the Cloud deployment mode, Team for Capella server is installed and administrated by Obeo, and Team for Capella clients are accessible through a remote desktop technology.
- **On-Premise**: With the On-Premise deployment mode, Team for Capella is installed and administrated on the client's infrastructure. The multiple ways to install/deploy Team for Capella are described in section Installation Architecture.

Applicability

This guideline is applicable to following versions of Capella and Team for Capella:

Value name	Value
AppChk	[INSTALLDIR]\capella\capella.exe;
Product Name	Capella x64 / Team for Capella x64
Product Version	7.1.1 / 7.1.1

This can be followed and adapted to the Linux version of Capella / Team for Capella. Path are

equivalents, the Linux bundle provides .sh scripts which are equivalent to the Windows .bat scripts described in this guide.

Document overview

This document is intended for persons in charge of installing Team for Capella.

It describes the nominal installation, configuration and uninstallation of Team for Capella.

Please refer to the Team for Capella documentation for more details and advanced configuration.

The **Team for Capella Guide** is available from the documentation embedded in the client, but also online, see Referenced Documentation.

To read the embedded documentation, launch a Team for Capella client (after installation), click on the *Help* menu in the top menu bar. Then click on the *Help Contents* button. A web page will appear with all documentation and especially the **Capella Guide** and **Team for Capella Guide** sections.

Chapter 2. Referenced documentation

Table 1. Reference Documents

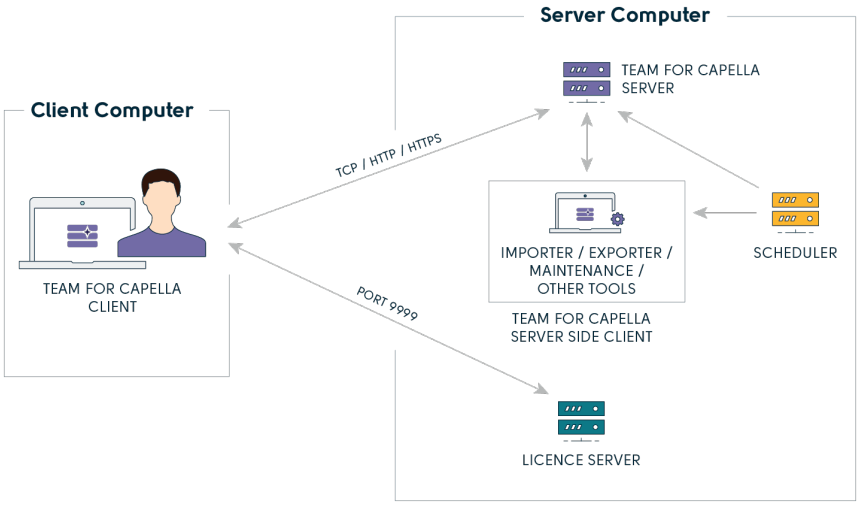
Title	Version
Team for Capella Guide: • Embedded documentation within Team for Capella • online documentation • PDF extracted from online documentation	7.1.1
REST API documentation of the Administration Server	7.1.1
Capella Online Installation Guide	7.1.1
Capella Release Notes	7.1.1
Team for Capella Release Notes	7.1.1

Chapter 3. Team for Capella Server installation

Installation Architecture

There are two main ways to install Team for Capella:

- **Local clients**:
  - server computer runs the Team for Capella Server and Scheduler,
  - n client computers run the Team for Capella Clients,

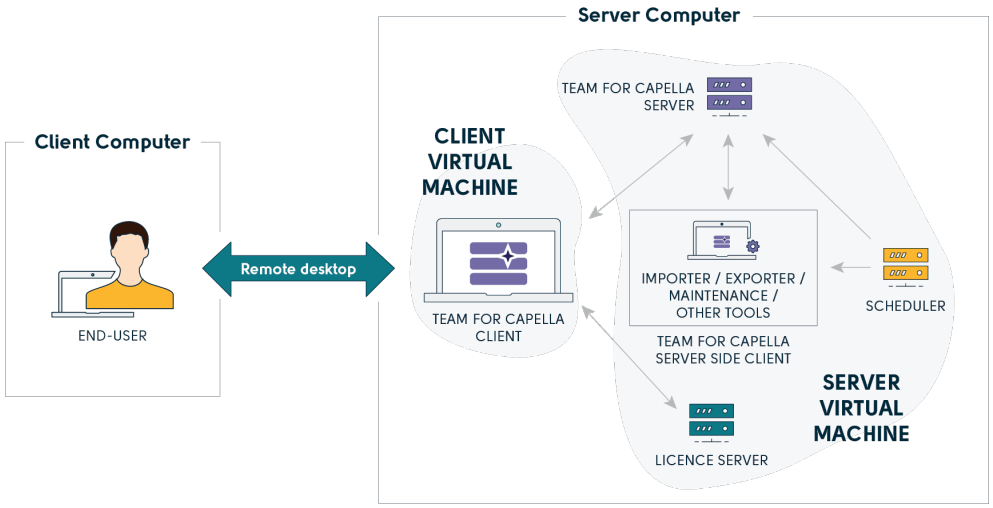


- **Remote clients**:

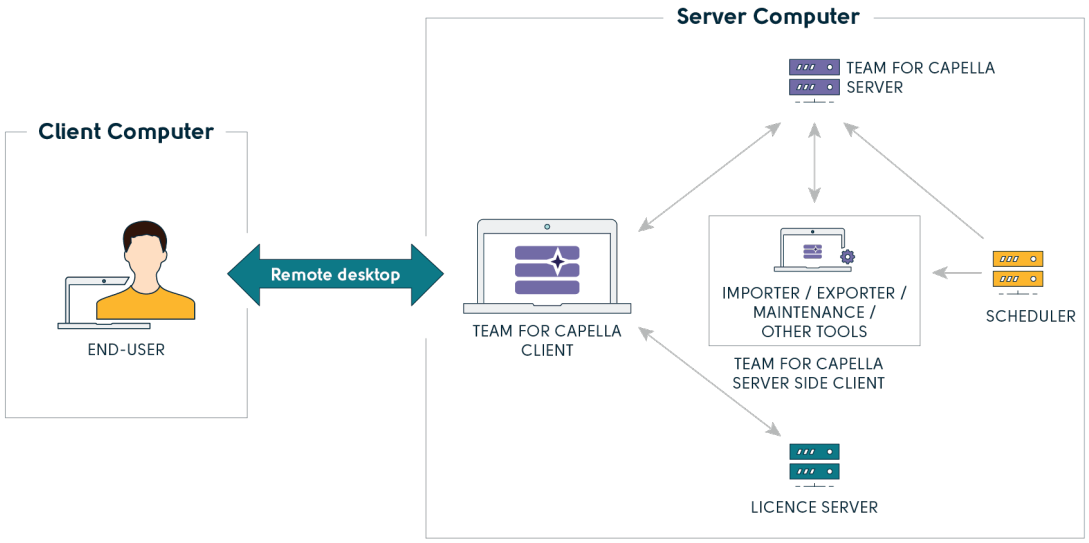
  The same computer runs the Team for Capella Server and Scheduler and n Team for Capella Clients (users connect to this computer using RDP). This allows to make the exchanges between the clients and the server independent of the network's bandwidth.

  In this configuration, the client and server can either be installed on the server computer or separated in two virtual machines. The latter is the recommended installation:

  - Client and server separated in two virtual machines:



- Client and server on the same server machine:



Recommended system requirements

Server computer recommended system requirements



If you installed a tool to act as a scheduler, it is mandatory to run it on the same computer as the Team for Capella Server.

For successful installation of Team for Capella Server, your computer must meet the following requirements:

- Local client installation (server side only):
  - 2 GHz processor,
  - RAM: 4 GB for the Team for Capella server + 3GB for the importer client
  - 15 GB of available hard disk space
- Remote client installation:

    - Multicore processor (2GHz)
        - 2 cores for Team for Capella Server, scheduler and license server
        - 1 core per running Team for Capella Client,
    - RAM:
        - 4 GB for the Team for Capella server + 3GB for the importer client
        - 3 GB per Team for Capella Client,
    - 15 GB of available hard disk space + 2 GB per Team for Capella Client,
    - 2 hard drives are recommended:
        - The first containing system files and software installation files (a SSD hard drive is mandatory if more than 8 users),
        - The second containing the Team for Capella Server file.
- System requirements:
    - Microsoft Windows 10/11 64 bits
        - no known issues with Windows 7/8
    - Microsoft Windows Server 2022/2025 64 bits
        - no known issues Windows Server 2008/2012/2016/2019
    - Compatibility with Open JDK 17, see Capella Online Installation Guide
        - Capella and Team for Capella are configured to use the JRE provided by Capella (Eclipse Adoptium Open JDK 21.0.11+10, JDK)
    - Jenkins LTS 2.580.2 installed as a service on the server computer, running with Java 21 minimum.
- Team for Capella database must be stored on a local hard drive,
- Security policies:
    - Virus scanner:
        - Team for Capella Server database files must not be scanned (*.db).
        - In addition, it should not be activated either on Capella models files: `*.aird, \*.capella, *.airdfragment, \*.capellafragment` and `\*.afm` ,
        - The license server hosts a collection of licenses stored in several encrypted `.ols` files. Those licenses pools must not be scanned.
        - Periodic analyses should not be launched when users are working (launch them at night),
    - Firewall:
        - Open the Team for Capella secure connection port (WSS and HTTPS, 8443 by default) and the license server port (9999 by default).
        - In addition, open the Scheduler port (8036 by default) and the license server monitoring port (8086) when these services must be accessed remotely.
        - TCP port 2036 and HTTP port 8080 are disabled by default. Open them when an

unsecured connection mode is explicitly configured, and restrict access to the intended systems.

- The computer should be fully dedicated to Team for Capella.

Client computer recommended system requirements (only for local client installation)

For successful installation of Team for Capella Client, your computer must meet the following requirements:

- 2 GHz processor
- 3 GB for Team for Capella client
- System requirements:
  - Microsoft Windows 10/11 64 bits
    - no known issues with Windows 7/8
  - Microsoft Windows Server 2022/2025 64 bits
    - no known issues Windows Server 2008/2012/2016/2019
- Compatibility with Java Runtime Environment 17 (Eclipse Adoptium Open 21.0.11+10 is provided by Capella)
- Security policies:
  - Virus scanner:
    - It should not be activated on Capella models files: `*.aird, \*.capella, *.airdfragment, \*.capellafragment` and `\*.afm`.

Deployment recommendations

Network

Latency: Client and Team Server

It is recommended to provide a network with the lowest possible latency between the client and the server: in the order of 1 to 10 ms for a round-trip.

Latency: Team server and DB server

It is strongly recommended that the Team server and the DB server are located on the same physical server as latency between the Team server and DB server will impact greatly the overall performances of the solution.As such the best performing deployment is achieved by using the H2 database in embedded mode with its `.db` database file located on the same disk than the Team server.

If there is a requirement on the database that prevents from using H2, make sure that the latency is as low as possible.

Network stability

VPN are not recommended (it is a latency factor) as well as other network elements that could drop connections which are more or less inactive. As such wireless connection are also not recommended as any loss of connectivity might lead to instability in the product and loss of data. However, if a network element of this kind is mandatory, an SSH tunnel could be used as a workaround to avoid client/server disconnections.

Server isolation

It is strongly discouraged to deploy the server on a public WAN. Team for Capella should be the only way to edit the information stored in the database.

Scalability and size of models

Scalability and performances are highly dependent on the design of the domain metamodel, the implementation of this metamodel and the Viewpoint Specification Models. The following figures are given with an Ecore model and the EcoreTools tooling which applies the Sirius best practices.

The minimum physical memory dedicated to the Team server is **4 GB** for a deployment where the expected model size is in the order of **300 000 model elements**. The heap memory available for the server should be increased to support bigger models: **8GB** should support **600 000 model elements**.

The memory usage of the clients will increase when the model which is shared among the clients grows as such these resources might need to be increased for larger models with **8GB** being expected for models with **600000 elements** (the exact value might vary depending on the amount of information each model element holds).

The latency of end user operations requiring a full analysis of the model increase as the model grow, this includes: opening and closing a project, deleting model elements and representations, launching a transformation or a code generation. Opening a project (and hence collecting the model from the network) might take **up to 1 min** for a model with **500K elements**.

Models having **1 000 000 model elements** are the considered the upper limit for a Collaborative Server deployment.

A given server is expected to be used by **10 to 20 users** simultaneously depending on their level of activity.

Disclaimer

Notwithstanding what was stated previously, Team for Capella product is not warranted to run without any error or interruption. Obeo does not make any warranty regarding the statements that are under the chapter Deployment recommendations, this chapter is provided for information purposes.

You acknowledge and accept the risks involved in using these products which could include without limitation, downtime, loss of connectivity or data, system crashes, bad performances or performance degradation.

Installation procedure

The date/time and the time zone of the server must be correct to make the scheduler work as expected.



To use existing models in a new version of Team for Capella, copies of these models have to be kept (in files format) before removing the old version.Once the new version is installed, the migration procedure will be performed on these models.

Team for Capella Server installation procedure

Installation

Preparation steps:

1. Download and extract the archive `TeamForCapella-7.1.1-win32.win32.x86_64.zip` in a given directory. It contains a `TeamForCapella` directory with 4 sub-folders and 1 file:
   - `lic-server`: contains a floating license server which allows several users to share the same product licenses.Each license can be used by only one user simultaneously.
   - `server`: the Team for Capella server
   - `tools`: contains some scripts, properties files and pre-configured jobs to configure a Jenkins as scheduler
   - `updateSite`: the Team For Capella update site for the client
   - `license.html`
2. Download the Capella 7.1.1 bundle from eclipse.org repository.

Client Installation

1. Unzip Capella bundle.
2. Move the capella and samples folders into the `TeamForCapella` directory.
3. Resulting structure of `TeamForCapella`
   - `capella`
   - `lic-server`
   - `samples`
   - `server`
   - `tools`
   - `updateSite`
4. Navigate to the tools folder and execute `installTeamForCapellaInCapella.bat`. The installation script will install the Team for Capella features in Capella, update the splash screen and update some properties in `capella.ini` and `config.ini`.

It is configured by default to retrieve the Team For Capella update site in the folder:

`TeamForCapella\updateSite`

The *repository* property can be updated in the script to reference it from another location.



This Team for Capella client (capella folder) should be used only for the Scheduler jobs: it must not be moved or renamed as its .exe files are referenced from the pre-configured jobs (Scheduler) and scripts (tools folder).

It can also be zipped and provided to user in case of local client installation, Team for Capella Client installation procedure.

In remote clients installation, you need to **copy the full** capella **folder** and **paste it** as *capella_client* in the same folder for example or **paste it** on the targeted client virtual machine. Then this client can be started on Windows Server and accessed with Remote Desktop. If you want to install additional functionality or add-ons, it will have to be done on both server-side and user-side clients.

Configure the default WSS/HTTPS connection

Team for Capella uses WSS for repository traffic and HTTPS for REST administration traffic on port 8443 by default. Before the first server start:

1. Obtain a PKCS#12 keystore (`.p12` or `.pfx`) containing the server private key and complete certificate chain. A certificate issued by a trusted private certificate authority is recommended.
2. Check that the certificate SAN contains the server FQDN for remote clients and `DNS:localhost` for importer and exporter callbacks on the server host. Include `IP:127.0.0.1` only if a connection explicitly uses that address.
3. Configure `server\configuration\fr.obeo.dsl.viewpoint.collab.server.admin\admin-server.properties` with the keystore path and passphrase. Prefer a system property or environment variable for the passphrase.
4. If the certificate cannot contain `DNS:localhost`, replace the local repository host and REST `httpHost` values used by the CLI tools and Jenkins jobs with the server FQDN contained in the SAN. Also replace the GET and POST `httpHost` values in `admin-server.properties`.
5. Distribute the issuing CA chain to clients when it is not already trusted. For a test-only self-signed certificate, follow Generate a self-signed PKCS#12 keystore for testing and Trust the WSS certificate in the Capella Java runtime.

Remote client defaults use the server FQDN, port 8443, and WSS. Packaged CLI tools and Jenkins jobs already use `localhost`, port 8443, WSS, and HTTPS. Keep these local defaults when the certificate SAN contains `DNS:localhost`; otherwise replace only their repository host and REST `httpHost` values with the server FQDN. See Default secure connections: WSS and HTTPS for the complete configuration and verification procedure.

Extensions installation

If meta-model extensions or add-ons are needed, use **one** of the following ways to install them:

- Use the Help > Install New Software... wizard if the extension are provided as update sites

- Otherwise if they are provided as dropins
  - Either unzip/copy their binary files in the folder `TeamForCapella\capella\dropins`
  - Or:
    - Unzip/copy them in any folder (it can be a shared folder between this server installation and client installations)
    - Modify the configuration file `TeamForCapella\capella\capella.ini` by adding the following parameter, **after -vmargs** :

```
-Dorg.eclipse.equinox.p2.reconciler.dropins.directory=<ExtensionFolder>
```



Exactly the same extensions have to be installed **on all clients** including the server-side client (`capella` and `capella_client`). Installation of a new version of an addon may require a migration, see Capella Addon migration.

Jenkins scheduler installation

Team for Capella provides applications to manage the CDO repositories with the shared projects.

These applications can be triggered from Capella, but Team for Capella also provides Jenkins jobs in order to manage them with a web interface.

See how to install Jenkins and our specific jobs in the documentation System Administrator Guide > Jenkins Installation.

If the target server cannot access the Internet, follow [installation-guide:jenkins_offline_installation_procedure:::].

Administration features installation

For system administrators, it is useful to install administration features such as the "Durable Lock Management" and "User Management" views and the "User Profiles" feature to manage users access rights.

Two ways to install these features in the Team for Capella Client:

- (recommended) Execute the script `TeamForCapella\tools\installAdminFeatures`
- install them manually from the Team for Capella client
  - "Help > Install New Software"
  - select the T4C update site and check the features under "Team for Capella - Administration".

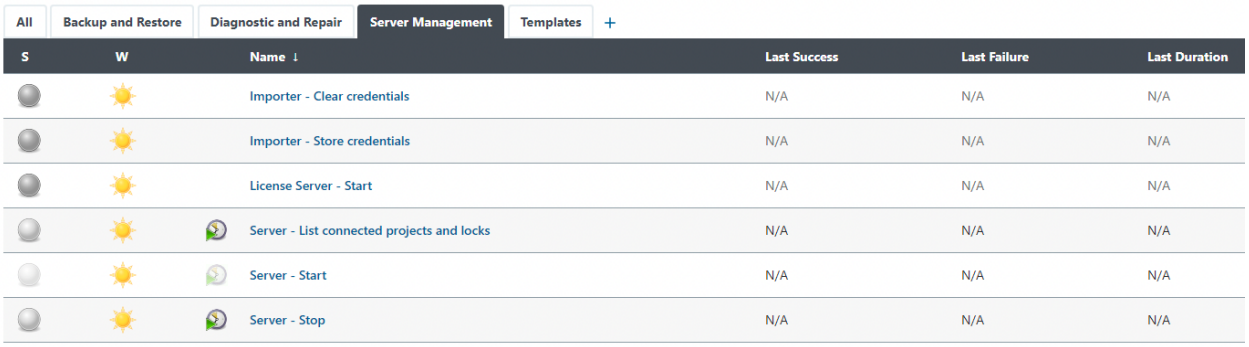
For more details about administration views and User Profiles, please refer to the documentation in System Administrator Guide > Server Administration > Administration Views and System Administrator Guide > Access Control (User Profiles).

License software installation

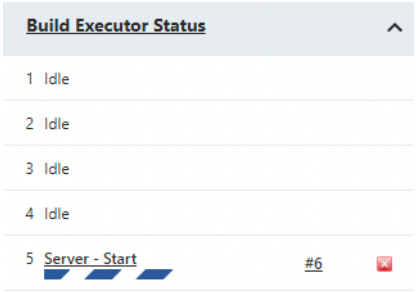
You need to install and configure the license server and then configure the Team for Capella client.

Installation verification

1. Connect to the scheduler admin page using the default URL http://localhost:8036,
2. Check that the job Server – Start has been automatically launched.



On the scheduler main page, the job Server - Run should appear the Build Executor Status section:



3. Once the server is started, it can be used to check the Team for Capella Client installations (see Team for Capella Client installation procedure),
4. Launch the `Server - List connected projects` and check its Console output,
5. Launch the `Database  backup` job from the Backup and Restore tab,
6. Stop the server using the `Server - Stop` job and the `Server - Run` job should be stopped.

Team for Capella server configuration

The Team for Capella Server is provided with a basic configuration. You can find all the documentation to configure it further in System Administrator Guide > Server Configuration. By default, the server authentication is based on a users file.

As a first step of server configuration, it is recommended to check the different authentication methods provided by Team for Capella, such as

- LDAP (subchapter System Administrator Guide > Server Configuration > Activate LDAP Authentication)
- and OpenIdConnect (subchapter System Administrator Guide > Server Configuration > Activate

OpenID Authentication).

WSS and HTTPS on port 8443 are the default connections. TCP and HTTP can also be enabled whenever required, including in parallel with WSS and HTTPS, but unsecured connections are not recommended for production; see Unsecured connection modes.

To manage your repository with its shared project, it is recommended to check the documentation of the Jenkins scheduler in Project Administrator Guide > Jenkins Configuration.

How to reuse a previous server configuration with a v7.1.1 server

When installing a new version of Team for Capella, some parts of the server/client configurations of the previous version can be reused.

Procedure

- Manually import your data or launch the backups jobs.
- Launch the `Server  Stop` job, terminate the `License Server - Run` job or end tasks `server.exe` and `lic_server.exe` in task manager (processes sheet)
- If you were using a Team for Capella version before 7.0.x, stop `TeamForCapellaScheduler` (service sheet) and remove it (see Uninstallation procedure).
- Keep the entire Team For Capella installation which contains all the files you may reuse.
- Choose among the following configuration you want to reuse

Server CDO

- Copy only the repository definitions required from `C:\xxx\T4C\server\configuration\cdo-server.xml`.
  - Do not blindly copy an old TCP acceptor. Keep the delivered WS/WSS acceptors for the default v7.1.1 transport.
  - the repository configurations can be reused except for *userManager*, *securityManager* tags.
- Configure the type of connection to the server:
  - For the default WSS/HTTPS connection, install the server certificate and update remote clients to the server FQDN and port 8443. Packaged CLI tools and Jenkins jobs already use `localhost`, port 8443, WSS, and HTTPS. Keep `localhost` when it is present in the certificate SAN; otherwise replace their repository host and REST `httpHost` values with the server FQDN. See Default secure connections: WSS and HTTPS.
  - Direct SSL remains available as a non-default alternative; see Direct SSL connection alternative.
  - If an unsecured connection is required, see Unsecured connection modes. Such a configuration is not recommended for production.
- if the server was configured with the authenticated configuration, the content of `C:\xxx\T4C\server\configuration\users.properties` can be kept as is.
- If the server was configured with the user profile configuration, the user profile model can be

reused. Follow the steps described in System Administrator Guide > Access Control > User Profiles > Import/Export User Profiles Model.

- You may also consider all properties or parameters you may have set in the `server.ini` file.
- If you are doing a new installation with the same version, you can reuse the database. Copy the `C:\xxx\T4C\server\eclipse\db` folder. Note that the database cannot be reused if the *Audit mode* activation changes between both installations.

Scheduler

- In previous versions (before 5.1.0, included), the scheduler was packaged inside Team for Capella, in the `C:\xxx\T4C\scheduler\jenkins_home` folder
  - `config.xml` contains the global Jenkins configuration which can be reused
  - `jobs\` contains the jobs definition with the build history. `jobs\<xxxjob>\config.xml` can be reused modulo some parameter changes that may be needed (for example for importer executable call).
- Since 5.2.0, Team for Capella does not pack a third party tool to manage the repository content (such as Jenkins). You will need to access its installation folder or *Home* directory.
- Customization made on those files/folders might be reported on your new custom installation made for this version.
- Check the modification done in each job with the **Job configuration history** available in each job.
- It is possible to create a v6.1 folder in the Scheduler interface, move the existing jobs in this folder, and check the Jenkins scheduler installation section to update Jenkins to the recommended version and install the new jobs. Then compare your v6.1 jobs with v7.0 jobs.

Tools

- You may consider all system properties or parameters you may have set in the **importer.bat** file **(or importer.ini for older versions)**. That information has to be transferred and adapted
  - From `C:\xxx\T4C14\capella\eclipse*importer.bat*` or `C:\xxx\T4C52\tools*importer.bat*`
  - To `C:\xxx\T4C\tools*importer.bat.*`
- Do the same for other scripts you have modified In `C:\xxx\T4C52\tools\\`: **`maintenance.bat`** ,**`command.bat`**.

Client (in remote clients installation)

- You may consider all properties or parameters you may have set in the `eclipse.ini` file. That information have to be transferred to `C:\xxx\T4C\capella_client\capella.ini.`
- If the IP address of the server did not change, you can keep the `OBEO_LICENSE_SERVER_CONFIGURATION` property system (just copy `-DOBEO_LICENSE_SERVER_CONFIGURATION=116168566473385529985114840…`.)

License server

- The license server version is now provided in the Team for Capella archive.
- The `*.ols` files may have been thought up to be reusable for your new Team For Capella version. In this case, you can keep your `\*.ols` files.

User Accounts Management

Several modes of access control can be used for each repository on the server:

- Identification (default mode): each user defined in the file `user.properties` is authorized to read and/or modify all models present on the repository.
- User Profiles: discriminating user rights are defined in a User Profiles model stored in the database.
- LDAP Authentication: this mode allows to authenticate with a LDAP server. It can be also used with authenticated or with user profiles.
- Not Authenticated Access: anyone can read and/or modify all models on the repository.

For the complete access control configuration documentation, refer to the documentation System Administrator Guide in chapters:

- System Administrator Guide > Access Control
- System Administrator Guide > Server Configuration
- System Administrator Guide > Server Configuration > Not Authenticated Configuration
- System Administrator Guide > Server Configuration > Authenticated Configuration
- System Administrator Guide > Server Configuration > Activate LDAP Authentication
- System Administrator Guide > Server Configuration > Activate OpenID Connect Authentication
- System Administrator Guide > Server Configuration > Activate WebSocket Connection
- System Administrator Guide > Server Configuration > Activate SSL Connection
- System Administrator Guide > Server Configuration > User Profile Configuration



In the current version, Team for Capella is configured with the “Identification” access control mode, ie. passwords are not encrypted. Refer to System Administrator Guide > Access Control if you need Authentication or Authorization mechanisms.

Uninstallation procedure

To uninstall Team for Capella:

1. Do a backup of projects stored on the Team for Capella Server,
2. Stop the server,
3. Stop the scheduler,

4. If a Windows Service was created, remove it:
    a. Using a Windows Command Line (as administrator), go to the scheduler directory,
    b. Execute the following command: `sc delete TeamForCapellaScheduler`.
5. If you installed a third party tool to manage your repository, such as Jenkins, uninstall it.
6. Remove the installation directory.

Chapter 4. Team for Capella Client installation

Requirements

Recommended system requirements

For successful installation of Team for Capella Client, your computer must meet the following requirements:

- 2 GHz processor
- 3 GB for (for big models, 4GB RAM)
- System requirements:
  - Microsoft Windows 10/11 64 bits
    - no known issues with Windows 7/8
  - Microsoft Windows Server 2022/2025 64 bits
    - no known issues Windows Server 2008/2012/2016/2019
- Compatibility with Java Runtime Environment 21 (Eclipse Adoptium Open JDK 21.0.11+10 is provided by Capella)
- Virus scanner should not be activated on Capella models files: `*.aird, *.capella, *.airdfragment, *.capellafragment and *.afm`.

Installation Procedure

Team for Capella Client installation procedure

Default client installation

You may take your Team For Capella client as it has been installed during the server installation. In that case, you need to make an archive of the capella folder and distribute it to users. Remember to include the samples folder if you want to ship the example model.

For Windows and Linux, it possible to follow the server installation procedure without the server nor Jenkins deployment and keep only the capella folder in the end.



Before archiving the Capella client that will be deployed on others computers, don't forget to remove the workspace used for the installation from the recent workspaces list (`Window > Preferences > General > Startup and Shutdown > Workspaces`).



It is also possible to manually install Team for Capella but note that the installation with the `tools/installTeamForCapellaInCapella.bat` is recommended.

Scripted client installation

It possible to use the installation script and target a Capella installation (Windows/Linux/macOS) It is similar to run the server server installation procedure without the server nor Jenkins deployment and keep only the capella folder in the end.

Preparation steps:

1. Download and extract the archive `TeamForCapella-7.1.1_ClientInstaller_Windows.zip` (or the `_LinuxAndMacOS.tar.gz` for other systems) in a given directory. It contains a `TeamForCapella` directory with 2 sub-folders and 1 file:
   - `tools`: contains some scripts, properties files and pre-configured jobs to configure a Jenkins as scheduler
   - `updateSite`: the Team For Capella update site for the client
   - `license.html`
2. Download the Capella 7.1.1 bundle from eclipse.org repository.

Client Installation

1. Install Capella (please refer to the Capella installation requirements).
2. Move `license.html` and the `tools`and `updateSite` folders next to the `capella` folder of your Capella installation directory.
3. Resulting structure of your Capella installation folder
   - `capella`
   - `samples`
   - `tools`
   - `updateSite`
   - `license.html`
4. Navigate to the tools folder and execute `installTeamForCapellaInCapella.bat` (`.sh` for other systems).
   - The installation script will install the Team for Capella features in Capella, update the splash screen and update some properties in `capella.ini` and `config.ini`.
   - It is configured by default to retrieve the Team For Capella update site in the folder: `..\updateSite`

Manual client installation

To manually install Team for Capella client in a Capella bundle (Windows/Linux/macOS), please follow the next steps:

1. Extract the archive `TeamForCapella-7.1.1_ClientInstallerOnly_Windows.zip` and keep only the `updateSite` and `tools` folders.
2. Download and extract the archive Capella 7.1.1 bundle (https://www.eclipse.org/capella/download.html)

3. Launch Capella (`capella.exe` for Windows, capella for Linux, Capella for macOS)
4. Click Menu *Help/Install New Software...* , add the archive file in the `updateSite` folder and select features in the *Team for Capella* category:
   - Team for Capella
   - User Interface to request a License
   - Other features from *Team for Capella* and *Team for Capella – Administration* categories are optional and not installed by the installation script.
5. Then you need to do the additional tasks performed by the installation script:
   - change the splashscreen:
     - replace the value of `osgi.splashPath` by `platform\:/base/plugins/com.thalesgroup.mde.melody.collab.ui` in `capella\configuration\config.ini` (`Capella.app/Contents/Eclipse/configuration/config.ini` for macOS)
   - copy the `license.html` file from `tools` to `capella` folder
   - avoid performance impact of CDO internal tracing:
     - add `-Dorg.eclipse.net4j.util.om.trace.disable=true` at the end of the `-vmargs` section in `capella\capella.ini` (`Capella.app/Contents/Eclipse/capella.ini` for macOS)
   - avoid third parties components logging noise
     - add `-Dlogback.configurationFile=configuration/logback.xml` at the end of the `-vmargs` section in `capella\capella.ini` (`Capella.app/Contents/Eclipse/capella.ini` for macOS)
     - copy the `logback.xml` file from `tools/resources/client_rootfiles` to `capella/configuration` folder (`Capella.app/Contents/Eclipse/configuration/` for macOS)
   - add possibility to override default repository connection information:
     - add `-pluginCustomization pluginCustomization.ini` before the `-vmargs` section in `capella\capella.ini` (`Capella.app/Contents/Eclipse/capella.ini for macOS`)
     - copy the `pluginCustomization.ini` file from `tools/resources/client_rootfiles` to `capella` folder (`Capella.app/Contents/Eclipse/` for macOS)
   - add the `license.html` in `capella` folder, copied from `tools\license`.
6. Add the license server configuration key as described in License software client configuration. If you have installed Team for Capella in a Capella bundle already completed with some add-ons, remember to check if they provide a “CDO Feature” for Team for Capella compatibility.

Steps 4 and 5 are done by the installation script.

Extensions installation

If meta-model extensions or add-ons are needed, use **one** of the following ways to install them:

- Use the Help > Install New Software... wizard if the extension are provided as update sites

- Otherwise if they are provided as dropins
  - Either unzip/copy their binary files in the folder `TeamForCapella\capella\dropins`
  - Or:
    - Unzip/copy them in any folder (it can be a shared folder between this server installation and client installations)
    - Modify the configuration file `TeamForCapella\capella\capella.ini` by adding the following parameter, **after -vmargs** :

```
-Dorg.eclipse.equinox.p2.reconciler.dropins.directory=<ExtensionFolder>
```



Exactly the same extensions have to be installed **on all clients** including the server-side client. Installation of a new version of an addon may require a migration, see Capella Addon migration.

Team for Capella Client configuration

For the complete client configuration documentation, refer to the chapter User Guide > Client Configuration > Preferences

1. Launch Team for Capella (`capella/capella.exe`),
2. Optional: clean the user's Secure Storage (it contains the save Login/Password, if "Remember Me" option was used): Go to menu **Window** > **Preferences** > **General** > **Security** > **Secure Storage**,
   a. Open the `Contents` tab,
   b. Select `[Default Secure Storage]`,
   c. Click on `Delete`,
   d. Upon request, restart the Team for Capella client.
3. Go to the menu **Window** > **Preferences** > **Sirius** > **Team collaboration**
   a. Set the server location to the FQDN contained in the server certificate SAN,
   b. Set port 8443 and select the WSS connection type,
   c. Click on `Apply`

The Java runtime used by the client is located in `capella\jre`. If the server uses a self-signed certificate, import its public certificate into this runtime's `cacerts` store as described in Trust the WSS certificate in the Capella Java runtime. Do not enable `trustall` in a production installation.

Multi-user client installation

When several operating-system users share one Capella installation, relocate the workspace and configuration areas to user-specific writable locations. Add the following options before `-vmargs` in `capella\capella.ini`:

```
-configuration
@user.home/TeamForCapella_7.1.1/capella/configuration
-data
@user.home/TeamForCapella_7.1.1/capella/workspace
```

Alternatively, equivalent `osgi.configuration.area`, `osgi.instance.area`, and `osgi.instance.area.default` system properties can be used after `-vmargs`.

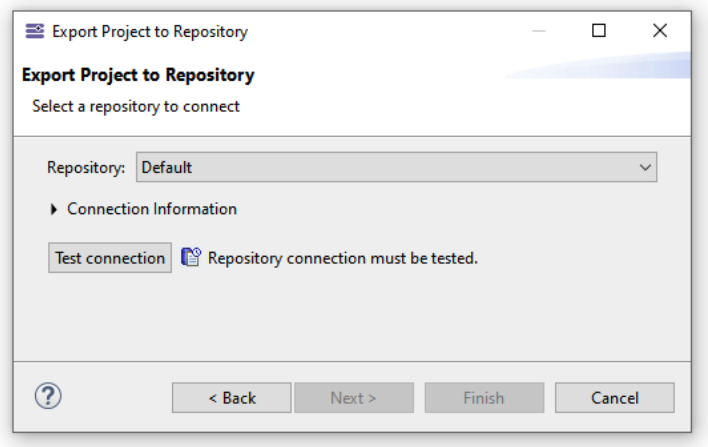
To share the registered repository list between users of the same installation, add:

```
-Dfr.obeo.dsl.viewpoint.collab.ui.registered.repositories.use.shared.configuration.location=true
```

The default value is `false`, so each user has a private registered-repository list. Ensure that all users can read the shared configuration and that each user can write to their own configuration and workspace.

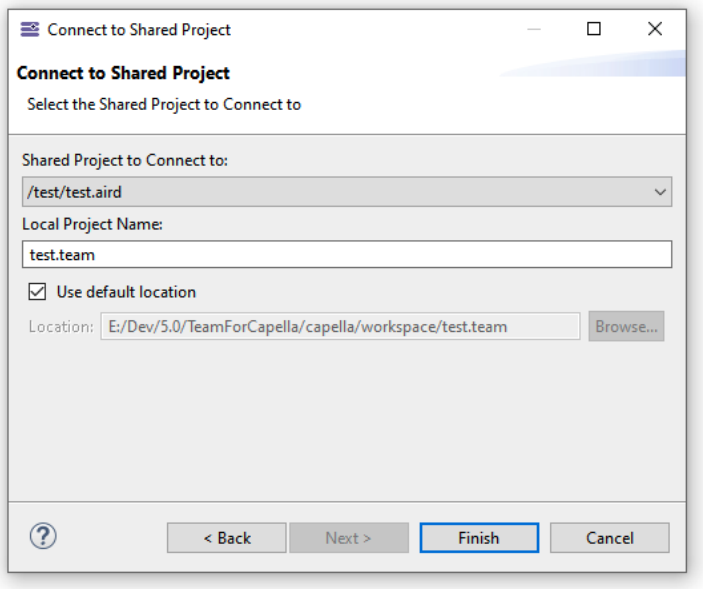
Verification installation procedure

1. **Create a new project > Right click in the Capella Explorer > New > Capella Project**
   a. Call it Test for example > Finish
2. Export the project to the remote repository
   a. Right click on the project > Export > Team for Capella > Export model to remote repository>Next,
   b. Expand *Connection Information* and verify the default server FQDN, port 8443, and WSS connection type,
   c. Click on Test Connection :



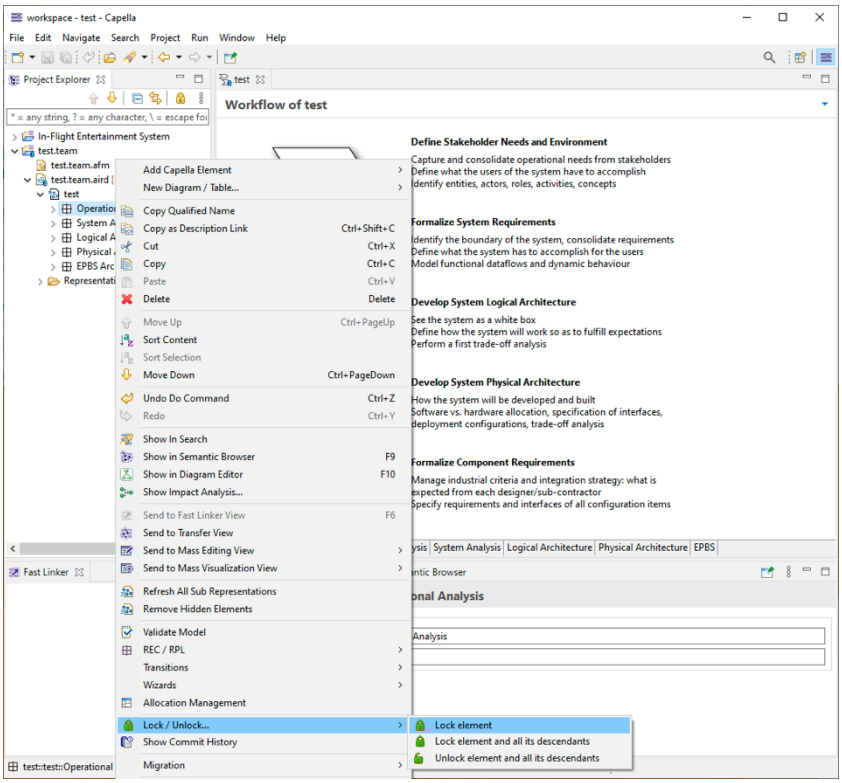
   d. Provide the user name and password (by default: user1/user1, user2/user2, user3/user3, admin/admin).
   e. Click on Finish.
3. Connect to the remote project previously exported,
   a. Right click in the Capella Explorer > New > Capella Connected Project
   b. Test Connection,

c. Select sthe Shared project from the list:



d. Click on Finish,

4. You should now be able to work on the project on the remote repository.



Chapter 5. Team for Capella license software installation

Server installation

1. The license server is provided in the `TeamForCapella-7.1.1-win32.win32.x86_64.zip` archive. After the preparation steps (see Team for Capella Server installation procedure), it is located in the folder `TeamForCapella\lic-server`
2. Unzip the `OLS.zip` archive in `TeamForCapella\lic-server\OLS`, the `OLS` folder should contain 4 `.ols` files.
3. You can choose to:
    - either launch the license server from the scheduler's job `License Server - Run` (disabled by default)
    - or directly launch the tool using the command `lic-server -keys ./OLS -verbose`

Additional parameters and documentation can be found in the `DOCUMENTATION.txt` file located in `TeamForCapella\lic-server\`.



The server configuration and more especially the `.ols` files should not be modified, moved or even accessed while the server is operating.Stop the license server before doing such operation.

Optional HTTPS monitoring endpoint

The license server monitoring endpoint is separate from the WSS/HTTPS connection used by the Team for Capella repository server. If organizational policy requires HTTPS monitoring, configure the `License Server - Run` job with a dedicated HTTPS port, for example:

```
lic-serverc.exe --launcher.ini lic-server.ini -verbose -consoleLog -keys ./OLS -port
9999 -httpPort 9443 -useHttps
```

Then reference a dedicated keystore in `lic-server\lic-server.ini` after `-vmargs`:

```
-Dorg.eclipse.jetty.ssl.keystore.path=../certs/<license-server-keystore.p12>
-Dorg.eclipse.jetty.ssl.password=<keystore-password>
-Dorg.eclipse.jetty.ssl.keystore.type=PKCS12
```

Protect the keystore and its passphrase according to the deployment security policy. This optional setting does not change the license checkout port encoded in the client connection key.

Client configuration

In order to connect Team for Capella instances with the license server, a connection key must be used to retrieve its address and port.

You should have received this key with the license server bundle and your .ols files. In order for Obeo to generate the key and send it to you, please provide the following information:

- preferably the DNS otherwise the IP Address of the machine that will host the licence server, seen from the machine where the Team for Capella client is installed. It means that if you have a WAF(Web Application Server) between the two machines then the DNS or IP Adress of the WAF should be provided.
- The associated port. By default it is 9999.

The `capella\capella.ini` file has to be modified with the following line to define the configuration of the server (after `-vmargs`):

```
-DOBEO_LICENSE_SERVER_CONFIGURATION=<connection key>
```

This must be done on all clients.

A license token is retrieved at the first connection attempt done by Team for Capella. It is then revoked when the last connected Capella project is closed or when Team for Capella closes. This license is verified from times to times with the server while you are using Team for Capella.

Clients are programmatically throttled and will not send requests to the server more frequently than once every minute. This throttling means some of the client actions might have a delay before being distributed to the server, for instance if one user stops using a given feature, 2 minutes of delay at most can be necessary for another user to be able to get the token.

The client/server communication is request based, no connection is kept alive for longer than just a request.

Chapter 6. Migration process

Version compatibility

Table 2. Version Compatibility (sorted by delivery dates)

Team for Capella	(based on) Capella	(third party software) Sirius
1.0.3 (32bits & 64bits)	1.0.3	Sirius 3.1.6
1.1.4 (32bits & 64bits)	1.1.4	Sirius 4.1.9
1.2.2 (64bits)	1.2.2	Sirius 5.1.4
1.3.2 (64bits)	1.3.2	Sirius 6.1.4
1.4.0 (64bits)	1.4.0	Sirius 6.3.0
1.4.1 (64bits)	1.4.1	Sirius 6.3.1
1.4.2 (64bits)	1.4.2	Sirius 6.3.3
5.0.0 (64bits)	5.0.0	Sirius 6.4.0
5.1.0 (64bits)	5.1.0	Sirius 6.5.0
5.2.0 (64bits)	5.2.0	Sirius 6.6.0
6.0.0 (64bits)	6.0.0	Sirius 7.0,1
6.1.0 (64bits)	6.1.0	Sirius 7.1.0
7.0.0 (64bits)	7.0.0	Sirius 7.4.1
7.0.1 (64bits)	7.0.1	Sirius 7.4.5
7.0.1_202509 (64bits)	7.0.1	Sirius 7.4.5
7.0.1_202601 (64bits)	7.0.1	Sirius 7.4.5
7.0.1_202607 (64bits)	7.0.1	Sirius 7.4.5
7.1.0 (64bits)	7.1.0	Sirius 7.4.14
7.1.1 (64bits)	7.1.1	Sirius 7.4.14

Model migration from previous version to v7.1.0

“**The CDO Framework cannot handle multiple version of the same metamodel on the same repository. Capella projects of different Capella versions needs to be exported on separate repositories**”.

To use a previous version model in Team for Capella v7.1.0, a migration must be done to be compliant with the new metamodel and file extensions.

This model migration is provided by Capella and must be done between each minor version (from v1.4.x to v5.x, for example).

The process to migrate a model to v7.1.0 from a shared repository follows the following steps:

1. If Team for Capella users have created local diagrams (i.e.: diagrams stored in the local `.aird` file), they have to move all diagrams they want to keep to a remote `.aird` or `.airdfragment` ("**Move Diagrams**" action),

2. Import locally the model to migrate ("**Import... / Team for Capella / Import model from remote repository**") using previous version (v5.x) of Team for Capella. Make sure that the associated team server is running before importing from remote repository. Another way is to use the last valid import of the Scheduler.

   → Make a baseline of the imported model,

3. Migrate the previously imported model ("**Migrate Project toward current version**") in a Team for Capella v7.1.0. (The migration can also be done in Capella v7.1.0).To do so, please refer to *Migration* dedicated chapter of Capella online installation guide,

   → Make a baseline of the migrated model,

4. Export the migrated model to the Team for Capella server v7.1.0 ("**Export... / Team for Capella / Capella Project to Remote**") using Team for Capella v7.1.0. Make sure that associated team server is running before exporting to remote repository.

After performing these steps, the model in the shared repository is in right version.Team for Capella has to be upgraded on client's computers.Then users can connect and work on the model.They do not need to do any migration.

Capella Addon migration

Some Capella addons extend the Capella metamodel with new metaclasses. Installing a new version of an addon may introduce a different version of the metamodel extension that alters existing metaclasses. In that case, a migration will be needed, Capella projects using these addons will need to be exported to a new repository. Therefore, the same procedure as the previous part is needed.

Model migration from an older version

The model migration is necessary between each version considering only minor and major version change.

- For the first model migration, you need to reproduce only steps **1**, **2** and **3** described above.
- For the following intermediary model migrations, you need to reproduce only step **3**.
- For the last model migration, you need to reproduce only steps **3** and **4**

For example, you start from v1.4.2 model

- do steps **1**, **2** and **3** with *previous version = 1.3.2* and new version = 1.4.2,
- do step **3** with *previous version = 1.4.2* and *new version = 5.2.0*
- do step **3** with *previous version = 5.2.0* and *new version = 6.1.0*
- do steps **3** and **4** with *new version = 7.0.0*.

Changes for Server Installation from v7.0.1_202607 to v7.1.0

Default connection changes

Team for Capella v7.1.0 uses WSS and HTTPS on port 8443 by default. Previous installations commonly used TCP on port 2036 for repository traffic and HTTP on port 8080 for REST administration traffic.

When upgrading:

1. Keep the `cdo-server.xml` delivered with v7.1.0 and copy only the required repository definitions from the previous file. Do not restore TCP as the default acceptor.
2. Provide a PKCS#12 keystore (`.p12` or `.pfx`) whose certificate SAN contains the server FQDN for remote clients and `DNS:localhost` for the local CLI tools and Jenkins jobs. Configure its path and passphrase in `admin-server.properties` before the first start.
3. Keep `admin.server.jetty.https.enabled=true`, `admin.server.jetty.net4j.enabled=true`, and port 8443.
4. Update firewalls and reverse proxies to allow WSS and HTTPS on port 8443. Ports 2036 and 8080 can be closed unless an explicitly restricted unsecured mode is required.
5. Update remote clients and customized client settings to use the server FQDN, port 8443, and connection type WSS. The delivered CLI tools and Jenkins jobs already use `localhost`, port 8443, WSS, and HTTPS. For copied or customized jobs and scripts, retain these secure protocol and port settings; if the certificate SAN cannot contain `DNS:localhost`, replace their repository host and REST `httpHost` values with the server FQDN.
6. If the certificate chain is not already trusted, install it in the `cacerts` store of the Java runtime embedded in each Capella client and in the server-side Capella runtime used by Jenkins jobs.
7. Test a client connection, an import or export, a database backup, and the server start/stop jobs before reopening the service to users.

For the complete secure setup, see Default secure connections: WSS and HTTPS. For an unsecured compatibility endpoint, see Unsecured connection modes; this configuration is not recommended for production.

Changes for Server Installation from v7.0.1_202509 to v7.0.1_202607

- DB files and dynamic repositories from v7.0.1 or v7.0.1_202509 can be reused in v7.0.1_202607.

Changes for Server Installation from v7.0.1 to v7.0.1_202509

Changes in Server Configuration

- DB files and dynamic repositories from v7.0.1 can be reused in v7.0.1_202509
- Repositories with OpenID Connect authentication
  - technicalUsers.properties files must be cleared: password are no more stored in clear text but encrypted with BCrypt.
  - Technical users can now be added/updated/removed without restarting the repository, via the REST api.
- REST Admin server realm passwords are now encrypted with BCrypt.
  - Secure storage must be cleaned from all Team for Capella keys: `fr.obeo.dsl.viewpoint.collab.credentials.*`
  - Before launching the server, delete `TeamForCapella/server/configuration/fr.obeo.dsl.viewpoint.collab.server.admin/secret.txt`
  - Credentials and passphrases can now be injected via environment variables instead of being stored in clear text in multiple configuration files. See available variables in Passing parameter using system property or environment.

Changes in Jenkins Jobs Installation/Configuration

- It is possible to do reuse current working jobs after updating the `TEAMFORCAPELLA_APP_HOME` if the new version has been installed at a different location

Changes for Server Installation from v7.0.0 to v7.0.1

Changes in Server Configuration

Changes in `cdoServer.xml`

- The default repository's data source is now named `repoCapella` instead of `capella` to match its repository name, ensure naming consistency and ease readability

Changes in Jenkins Jobs Installation/Configuration

- A new option has been added to the `Repository - Start` job to allow the re-initalization of a repository without having to restart the server nor to access the files.
- URLs of Jenkins plugins have been replaced in installation scripts in order to avoid redirect errors during installation on Linux.

Changes for Server Installation from v6.1 to v7.0.0

Changes in Server Configuration

Changes in `admin-server.properties`

- The property `admin.server.api.project.get.impl.config.XMLImportFilePath` in v6.1 becomes

`admin.server.api.project.get.impl.config.importFilePath` in v7.0.0

Changes in `cdoServer.xml`

- The websocket configuration `<acceptor type="ws" listenAddr="YourAcceptorName" />` in v6.1 becomes `<acceptor type="ws" name="YourAcceptorName" />` in v7.0.0

Changes in Jenkins Jobs Installation/Configuration

The recommended Jenkins version is now 2.440.3 LTS for the Team for Capella 7.0.0 and the required plugins list has changed. You can read more about the jenkins installation in section System Administrator Guide > Jenkins Installation of the user manual.

The configuration of the Jenkins jobs installation is no longer directly in the installation script `install-TeamForCapellaAppsOnJenkins.bat`/`install-TeamForCapellaAppsOnJenkins.sh`. In v7.0.0, the configuration is now in a separate file `install-TeamForCapellaAppsOnJenkins.properties`. You can read more about the Jenkins jobs installation in section System Administrator Guide > Jenkins Installation > Install Jenkins plugins and jobs required for Team for Capella of the user manual.

Changes in Jenkins Jobs

The following jobs have been renamed:

- `License Server - Start` in v6.1 becomes `License Server - Run` in v7.0.0
- `Server - Start` in v6.1 becomes `Server - Run` in v7.0.0
- `Projects - Import` in v6.1 becomes `Projects - Import - repoCapella` in v7.0.0

The following jobs have been added:

- `Repository - Commit history`, more information in the section Project Administrator Guide > Jenkins Configuration > Team for Capella Scheduler > Backup and Restore > Repository — Commit history of the user manual
- `Repository - Import projects from history`, more information in the section Project Administrator Guide > Jenkins Configuration > Team for Capella Scheduler > Backup and Restore > Repository — Import projects from history of the user manual
- `Repository - List projects`, more information in the section Project Administrator Guide > Jenkins Configuration > Team for Capella Scheduler > Backup and Restore > Repository — List project of the user manual

A new parameter was added to select the repository for the following jobs:

- All `Repository` Jobs
- All `Tools` Jobs
- `Projects - Export`

Other Changes

In the importer application:

- The parameter `-XMLImportFilePath` in v6.1 becomes `-importFilePath` in v7.0.0
- The parameter `-archivefolder <folder>` in v6.1 is replaced by two parameters `-archiveProject true -outputFolder <folder>` in v7.0.0

Chapter 7. Performance consideration feedback

This chapter will provide you some useful information about the performance you should expect and about performance on production context (with big sized model and many users)

Reference time

You may wonder, if the performances you have are normal, that is, if the time elapsed, when doing usual action in Team for Capella, is the right one compared to what should be expected.

Ensuring that the elapsed time doing your actions is the expected one, will help you to ensure that there are no issues that could come from other cause like network lag, lack of memory, antivirus process etc..

Case Studies

Some studies have been done on reference Capella models:

- The open source *IFE* model packaged with Capella
- *Combined IFEs*, a modified version of *IFE* which is artificially swollen to have an important sized model.

The next figure presents some characteristics on those models and the expressions used to compute them.

Table 3. Reference models

Characteristics	**Project**		
	IFE	**Combined IFEs**	
File (size in KB)			
Project	19 179	203 310	
.capella	1 475	13 464	
.aird	17 705	189 846	
Model (number of elements)			**AQL expressions to use in Interpreter view**
Capella elements	5 378	49 389	aql:self.eAllContents()→size() *select the first child of the .aird file in the Project Explorer*

Characteristics	Project		
Components	110	1 230	aql:self.eAllContents(cs::Component)→size() *select the first child of the .aird file in the Project Explorer*
Functions	224	2 652	aql:self.eAllContents(fa::AbstractFunction)→size() *select the first child of the .aird file in the Project Explorer*
Diagrams	111	880	aql:self.eResource().getContents()→filter(diagram::DDiagram)→size() *select a diagram in the Project Explorer*
Tables	1	8	aql:self.eResource().getContents()→filter(table::DTable)→size() *select a diagram in the Project Explorer*
Elements displayed in all representations	6 384	70 850	aql:self.eResource().getContents()→filter(viewpoint::DRepresentation).eAllContents(viewpoint::DRepresentationElement)→size() *select a Diagram in the Project Explorer (only for local projects)*

Refer to How to get characteristics of your model to see how to get the characteristics of your own model in order to compare it with those reference models.

Measures

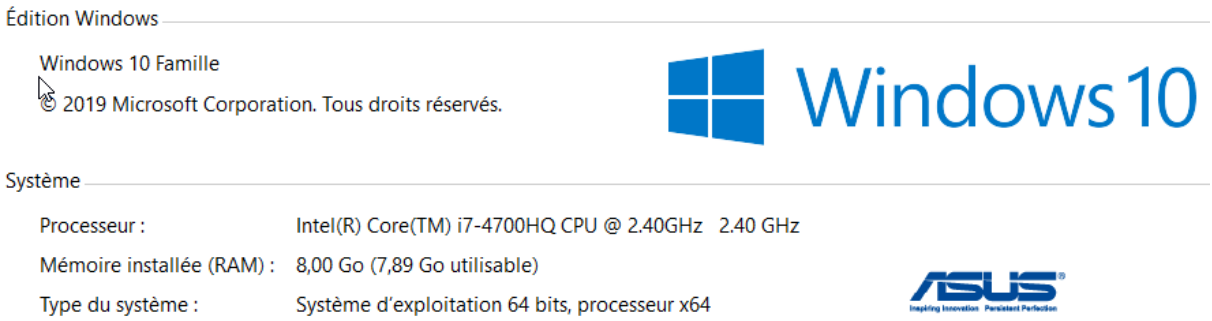
Measures contained in this document give some idea and order of magnitude of the performances expected for comparable environments.

They have been taken on two different deployments:

- *Local*: single computer with both client and server - IFE and Combined IFEs
- *OVH*: two virtual machines, one for the client and one for the server as recommenced in Deployment recommendations.

Computer used for the tests

Local Windows computer



OVH Windows computer



Reference elapsed time

The memory allowed to the Team for Capella client can be configured with the *capella.ini* file beside *capella.exe* executable.

Table 4. Measured time on test models (5.0.0)

Elapsed time (s)	IFE	Combined IFEs
	Xmx 3GB	Xmx 3GB
Export To Server	28,7	114
Open Session (first time)	3,7	9
Test Close Session	1	3
Open Session	2	6,5
Open the diagram (1)	4	6
Create a LFBD diagram on Root Logical Function. (2)	3	6
Save the session after steps from (3)	2	7

Analysis and conclusion

The majority of the scenarios could be done with the minimum recommended memory of 3Go.

Nevertheless, with model of big size (*Combined IFEs*), we need more RAM to work with constant performance for some actions like *Refresh all representations* or *Validate* on the whole model.

If the *Team for Capella* Client reaches, at execution, the maximum allowed memory, the performance may drastically fall particularly for actions that consume a lot of memory such as:

- Import a Capella project locally from the server
- Do the V*alidation* of the whole project
- Start *Refresh All Representation* action
- Save if many objects have been changed or deleted
- Some particular Capella functionality such as *functional transition*

Dedicated tests on *Combined IFEs*

The aim of this test session was to ensure that Team for Capella is able to have to keep the right performances after having used for a while.

- *Combined IFEs* is used as the Capella Project. It is a large sized project.
- 20 users are connected to the Capella project on the server
- 8 users are really working on the model. During the test, they will do standard actions such as
  - Open many diagrams
  - Create, Refresh, modify diagrams
  - Make functional transition for some of them
  - Regularly create semantic element through diagram or not
  - Use tool in diagram palette
  - Use the *Activity Explorer,* F8 and F9 to navigate
  - Validate some parts of the model

But also, other actions that demands significant amount of RAM

- The tests have been done with 21 virtual machines hosted on OVH cloud (20 clients and 1 server):
  - Client Xmx=3Go
  - Server Xmx=4Go

Server

Throughout the test, the server never reaches its maximum allowed memory of 4Go RAM and not even 3Go. It keeps constant performance.

Client

The client behaves constantly with good performance throughout the tests.

Nevertheless, the client may reach 3Go of memory if high memory consumption actions are performed such as:

- Import a Capella project locally from the server
- Do the *Validation* of the whole project
- Start *Refresh All Representation* action

Once the maximum allowed memory is reached:

- The user keeps having good performance for standard actions
- But the performance may fall drastically for high memory consuming actions. For that cases, the performance issue can be solved
  - using more allowed memory for the client (Xmx=4 or more)
  - simply restarting the client

Note that the potential performance issue encountered by one user has no impact on another user

conclusion

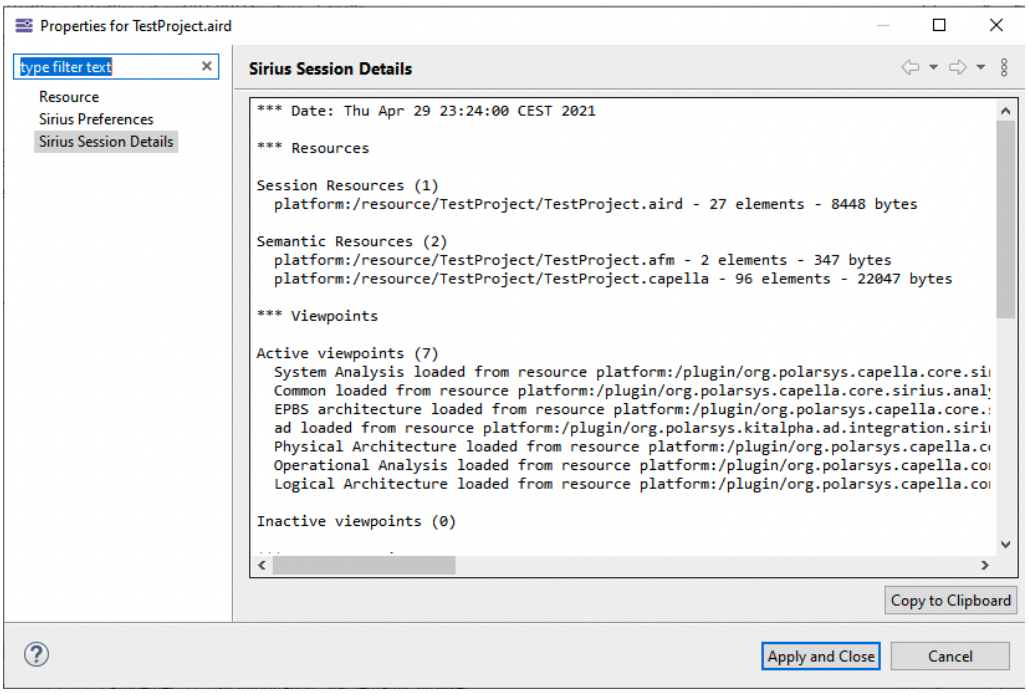
The model used is what we consider as a medium-big sized model.

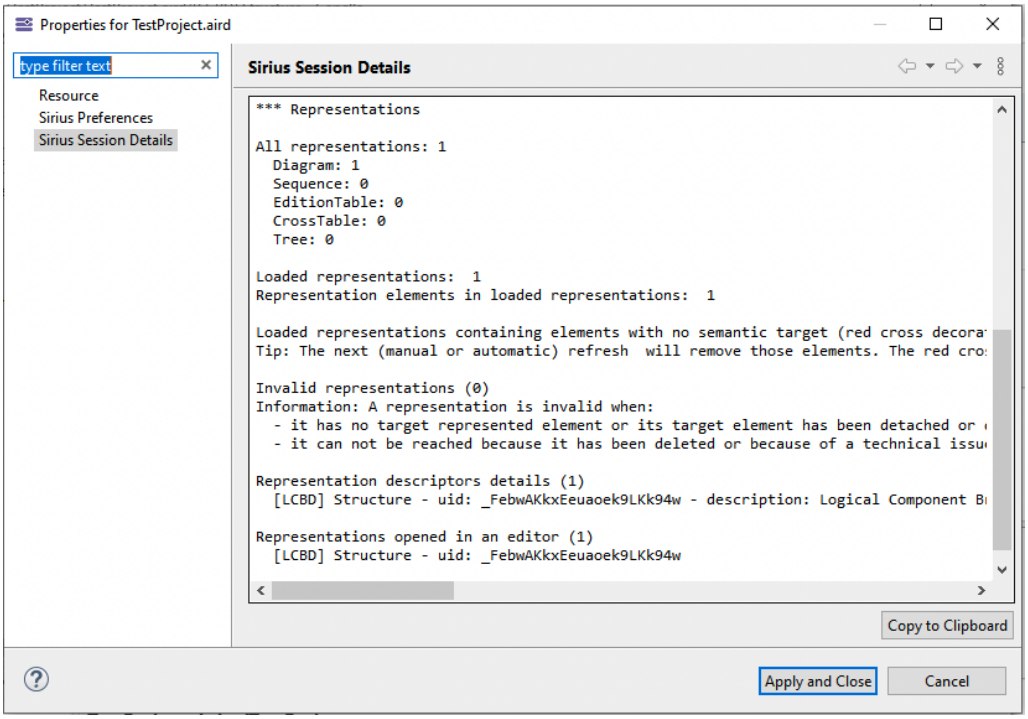
20 users are working on it.

- The server 4Go is well fitted for the use
- The client 3Go RAM is sufficient for most cases.It can be increased to have best performance when the user uses high memory consuming action

How to get characteristics of your model

The simplest way to get characteristics of your model is to use the *Sirius Session Details* tab available in the *Properties* contextual menu of the .aird file of your project:

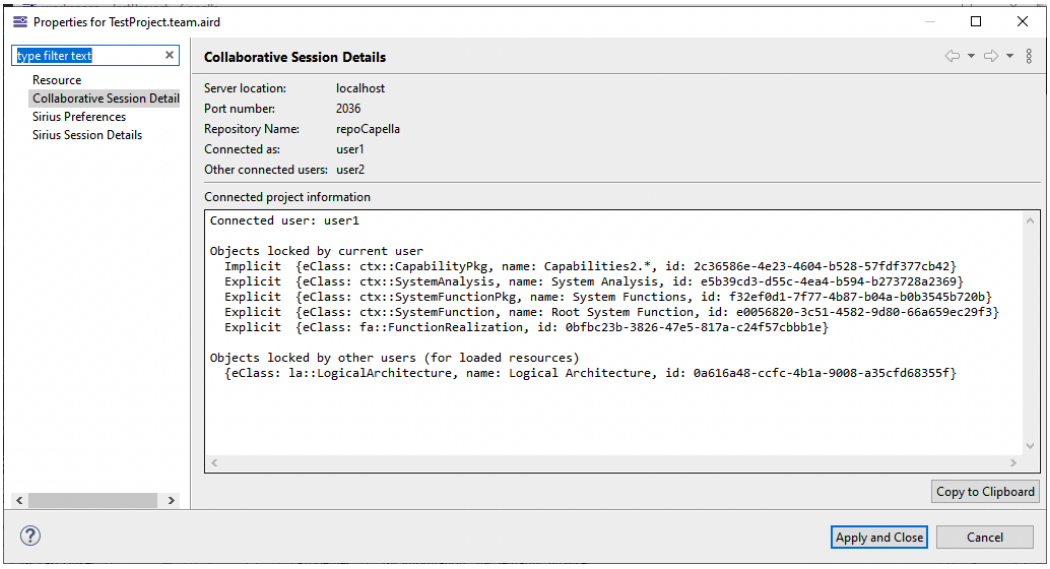




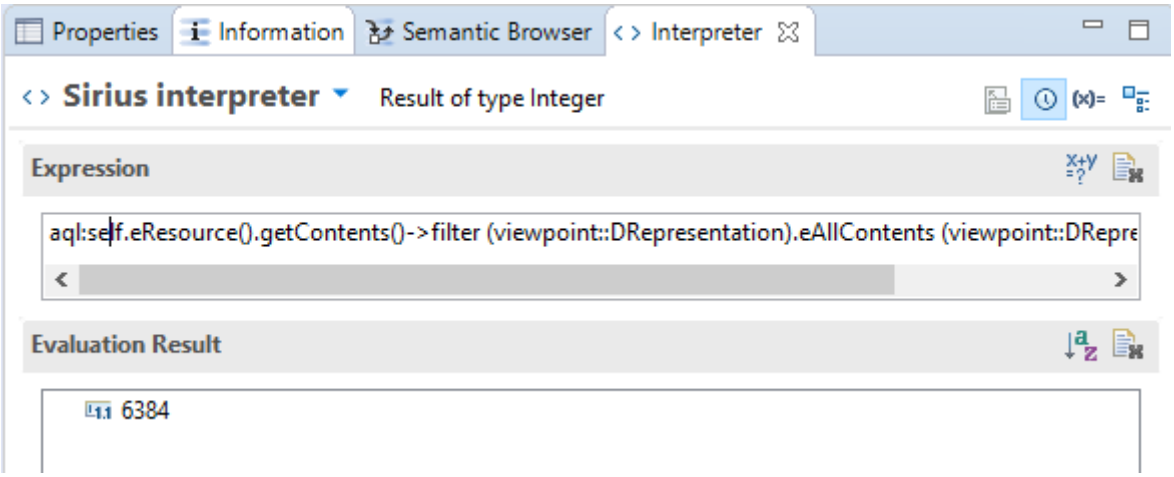
It will give you the following information about your project:

- Number, name, path and number of contained elements of each resources
- Selected Viewpoints
- Number of representations
- Some details about each representation (name type, id, status)
- Number of loaded representations
- Number of representations opened in editor

On shared projects, the *Collaborative Session Details* tab gives information about connected users and locks:

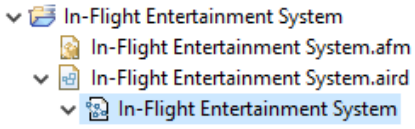


It is also possible to additional model characteristics thanks to the *Interpreter* view:



To get the characteristics of the *.capella* semantic resource :

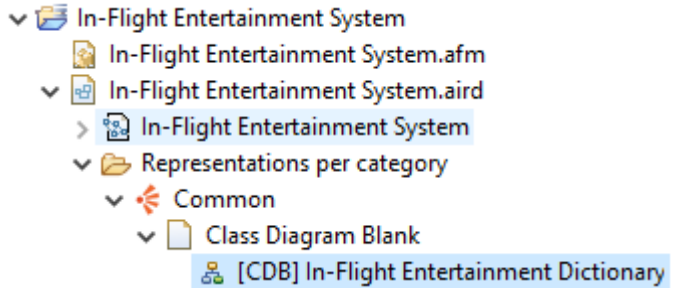
- select any semantic element



- write the AQL query in the *Interpreter* view to get the result in the lower part of the view.

To get the characteristics of the representations in the *.aird*

- select any representation



- write the AQL query in the *Interpreter* view to get the result in the lower part of the view